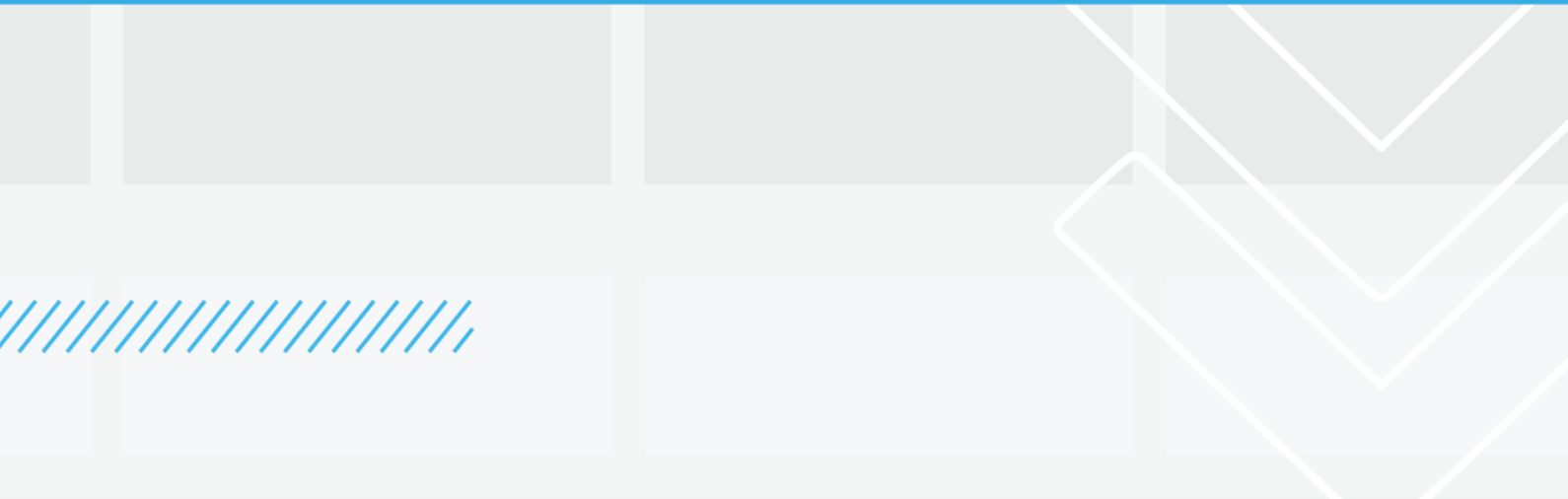





INICIATIVA PROGRAM.AR DE LA FUNDACIÓN SADOSKY

» Propuesta curricular para la inclusión de las Ciencias de la Computación en la educación obligatoria de la Argentina



Autoría (por orden alfabético):
Dabbah, Julián
Garzón, Magdalena
Gómez, Marcos
Martínez, María Cecilia
Martínez López, Pablo E. "Fidel"

Con la colaboración de:
Borchardt, Mara y Labhart, Mariana

Fundación
SADOSKY

<Program.AR/>
DIEZ AÑOS

Versión 04, julio 2023



Agradecemos a las y los docentes de educación primaria, secundaria y superior, a las y los integrantes de la iniciativa Programa.AR por su lectura crítica y valiosos aportes para el desarrollo de esta propuesta.

Editor: Fundación Sadosky

La versión digital de este documento se encuentra en

<https://curriculum.program.ar/>

Cómo citar este documento: Dabbah J., Garzón M., Gómez M., Martínez M.C., Martínez López P. E. *Propuesta curricular para la inclusión de las Ciencias de la Computación en la educación obligatoria de Argentina*. Fundación Sadosky, Buenos Aires, julio 2023.

Esta obra está licenciada bajo la Licencia Creative Commons

Atribución-NoComercial-SinDerivar 4.0 Internacional. (CC

BY-NC-ND 4.0) Para ver una copia de esta licencia, visitá

<https://creativecommons.org/licenses/by-nc-nd/4.0/deed.es>

Presentación

Esta propuesta curricular para la inclusión de las Ciencias de la Computación (CC) al currículo escolar¹ es el producto de las reflexiones sostenidas por el equipo de la Iniciativa Program.AR de la Fundación Sadosky e investigadoras e investigadores del campo de la Didáctica de las Ciencias de la Computación y las Ciencias de la Educación de distintas Universidades Nacionales a lo largo de los últimos 10 años.

En este documento se reúnen los saberes de CC que, fruto de estas reflexiones, se consideran necesarios para ejercer una ciudadanía plena en una sociedad organizada por la tecnología computacional. La apropiación de estos saberes por parte de las y los estudiantes implica poner en juego de manera conjunta prácticas computacionales y contenidos disciplinares enmarcados en propuestas de enseñanza significativas para la comprensión del mundo contemporáneo.

Es desde este enfoque que la Iniciativa Program.AR viene asesorando en la actualización del espacio de Educación Tecnológica a las provincias de Neuquén, Córdoba, La Pampa, Santa Fe, La Rioja y otros países como Uruguay.

En la primera parte del documento, se abordan los fundamentos de la propuesta curricular y se explicitan los criterios de su diseño. En la segunda parte, se especifican los saberes y alcances propuestos para el segundo ciclo de nivel primario y el ciclo básico y orientado del nivel secundario. La preocupación por estos ciclos expresa la decisión de priorizar, en ese momento de la escolaridad, las CC como objeto de estudio (es decir, que las y los estudiantes construyan saberes sobre el funcionamiento de los artefactos y los dispositivos y el rol de las tecnologías computacionales en la estructuración de vínculos sociales, la construcción de poder político y económico, etc.), mientras que para los ciclos anteriores, se recomienda centrarse en la alfabetización digital vinculada al uso de los artefactos y los dispositivos computacionales.

Se propone una estructura curricular integrada por áreas y ejes temáticos con dos propósitos: (I) proponer una terminología precisa que permita establecer en la escuela un lenguaje común para nombrar conceptos provenientes de las CC y (II) reflejar el modo en que estos conceptos se definen, relacionan y complejizan en la disciplina que permita dar cuenta del espiralamiento y la profundidad de los alcances por ciclo. Además, se definen cuatro prácticas computacionales para reflejar los “modos de hacer” de la disciplina.

¹ Se considera que los currículos son recortes de un campo disciplinar para ser enseñados que luego las escuelas y el cuerpo docente van reconfigurando y adaptando según sus propios saberes, los contextos sociales y políticos en donde la escuela está inserta, las condiciones institucionales y las características de su alumnado (Terigi, 1999).

Al final del documento se presenta un glosario que precisa algunos de los términos mencionados en la propuesta.

Para la elaboración de este documento se realizó la revisión bibliográfica de los siguientes trabajos: *K-12 Computer Science Framework* (2016), *Computer Science Curricular Guidance for Associate-Degree Transfer Programs* (2017), *Guide to Teaching Computer Science* (2014), "It is time for more critical CS education" (2020), *Computer Science: A curriculum for Schools* (2012), *CSTA Standards* (2017), *ISTE Standards for Students* (2016); *Exploring Computer Science* (2015), *Núcleos de aprendizaje prioritarios de Educación Tecnológica* (2011), *Núcleos de Aprendizaje Prioritarios para Educación Digital, Programación y Robótica* (2018).²

Mara Borchardt

Directora Iniciativa Program.AR

Fundación Sadosky

² Ver fuentes en la sección Bibliografía.

Índice

Presentación.....	3
Índice.....	5
Primera parte. Fundamentos.....	6
La relevancia de los saberes de las Ciencias de la Computación en el mundo contemporáneo.....	6
Qué conocimientos engloban las Ciencias de la Computación.....	7
Por qué la escuela debe ser el lugar privilegiado para socializar estos saberes.....	9
El abordaje de las Ciencias de la Computación en el currículo argentino.....	11
Ejemplos de implementación en algunas jurisdicciones.....	12
Problemas que complejizan la inclusión de las CC en la educación obligatoria.....	13
Propósitos y enfoque de la propuesta curricular.....	14
Criterios para la selección de los contenidos.....	16
Segunda parte. Propuesta curricular.....	18
Áreas, ejes y contenidos.....	20
A. Área Ciudadanía y computación.....	23
Eje A1. Identidad digital.....	23
Eje A2. Estrategias para un uso seguro de internet.....	32
Eje A3. Computación y sociedad.....	35
B. Área Programación.....	46
Eje B1. Soluciones a problemas computacionales.....	46
Eje B2. Representación de información en la resolución de problemas computacionales.....	54
Eje B3. Lenguajes de programación.....	60
C. Área Infraestructura tecnológica.....	71
Eje C1. Organización y arquitectura de computadoras.....	72
Eje C2. Sistemas operativos.....	78
Eje C3. Redes e internet.....	82
D. Datos e inteligencia artificial.....	90
Eje D1. Recolección y modelado de datos.....	90
Eje D2. Ciencia de datos, inteligencia artificial y aprendizaje automático.....	94
Prácticas computacionales.....	103
A. Reconocer y desarrollar soluciones para problemas computacionales.....	103
B. Crear y/o modificar artefactos computacionales.....	103
C. Desarrollar, utilizar y abrir abstracciones.....	104
D. Identificar y comunicar problemas de dispositivos y artefactos computacionales.....	105
Glosario de términos.....	106
Referencias bibliográficas.....	115

Primera parte. Fundamentos

La relevancia de los saberes de las Ciencias de la Computación en el mundo contemporáneo

La vida contemporánea se organiza cada vez más en torno a [artefactos](#) y [dispositivos computacionales](#). Las tarjetas de pago del transporte público, las compras mediante aplicaciones del celular, las plataformas de series y películas, los sistemas de arbitraje para los deportes de competencia, los de seguridad por reconocimiento facial... son todos deudores de la [tecnología computacional](#)³.

Es tan habitual su uso que se ha popularizado la idea de que usarla y entenderla son equivalentes; sin embargo, no lo son. Cuando se hace referencia a lo que las personas “saben o no de tecnología”, muchas veces se confunde la capacidad de usar y consumir programas, aplicaciones y computadoras, con el conocimiento específico del funcionamiento de tales artefactos y dispositivos computacionales. Esto lleva a pensar que personas (tanto adultos como también niñas, niños y adolescentes) con conocimientos sobre la utilización de redes sociales, editores de textos y videos o búsquedas en Internet no necesitan estudiar computación. Es decir, no les haría falta saber de programación, arquitectura de computadoras, ni de sistemas operativos o de la estructura de redes e internet. Sin embargo, la falta de estos saberes limita de manera creciente y significativa el empleo crítico de estas herramientas e impide su transformación y creación.

Sin ir más lejos en el tiempo, el aislamiento obligatorio que se dispuso durante la pandemia del Covid-19 entre 2019 y 2021 aceleró los procesos de virtualización de actividades cotidianas y laborales. Esto evidenció las dificultades de algunas personas para comprender funcionamientos inesperados. Habitualmente presentaban dificultades para poder reconocer, describir, diferenciar e identificar en términos generales los problemas y saber si correspondían al *hardware*, al *software*, a la conexión a internet o al servicio de conectividad. Pocas contaban con el conocimiento necesario para comparar y analizar distintas opciones e implementar alternativas para resolver este tipo de problemas.

Asimismo, la ciudadanía debatió intensamente sobre el uso de plataformas sanitarias que operan con algoritmos de macrodatos para trazabilidad y geolocalización, con capacidad de detección y prevención temprana de casos de Covid-19 (Harari, 2020). Según la Organización Panamericana de la Salud, estos

³ En el glosario pueden encontrarse los argumentos que precisan la decisión de utilizar el término “computacional” en lugar de “digital”.

sistemas fueron clave para estimar casos y asignar recursos necesarios como camas de terapia intensiva y respiradores. Hubo muchos cuestionamientos sobre el uso de datos privados por parte del sector público (ANCCOM, 2020) lo cual resulta paradójico considerando el escaso cuestionamiento de la utilización de esos datos con fines comerciales por parte de las corporaciones.

En otro orden, los ciberdelitos crecieron un 300 % durante la pandemia. En Argentina, se registraron incrementos considerables en la utilización de *homebanking*, *mobile banking*, pagos remotos y comercio electrónico. Se tornaron cotidianos también los trámites remotos ante la Administración Pública Nacional a través de la plataforma de Trámites a Distancia (TAD). En este contexto, diversos reportes nacionales dan cuenta de un crecimiento exponencial de la ciberdelincuencia, proporcional al volumen de maniobras ligadas a la informática que tienen lugar. Entre las modalidades delictivas relevadas se incluyen: fraude, *phishing*, acceso ilegítimo, usurpación de identidad, *ransomware*, acoso y acceso indebido a cuentas bancarias. Un gran porcentaje de estos ataques afectaron a jóvenes a través de delitos de extorsión y acoso sexual (UFECI, 2020).

No obstante, de la brecha de saberes computacionales se ha hablado bastante menos tanto en los debates públicos como en los discursos pedagógicos especializados que marcan con énfasis las brechas de acceso y uso (Terigi, 2020).

Qué conocimientos engloban las Ciencias de la Computación

Ciencias de la Computación (CC) es el nombre que recibe la disciplina que estudia las bases y los fundamentos sobre los que se articula la creación y utilización de las [tecnologías computacionales](#). Se ocupa de los saberes necesarios para poder **formular soluciones efectivas y sistemáticas** a diversos tipos de problemas computacionales. Entre ellos y sin ser exhaustivos, retomando a Schapachnik y Bonello (2022), podemos mencionar:

- **La algoritmia.** Reúne los conocimientos necesarios para formular soluciones efectivas y sistemáticas ante diversos tipos de problemas. Por ejemplo, pensemos en un sistema de posicionamiento global (GPS, por su escritura en inglés) y la necesidad de sugerir a un usuario un camino entre todos los posibles según las condiciones de tránsito de un determinado momento.
- **La programación.** Nuclea los conocimientos necesarios para poder volcar esas soluciones algorítmicas a los diversos lenguajes que utilizan las computadoras. Muchas veces, y en particular cuando se habla de “llevar la programación a la escuela”, se engloba a la algoritmia dentro de la programación.

- **Las estructuras de datos y las bases de datos.** Son dos áreas temáticas que se encargan de la forma de almacenar la información de manera que pueda ser recuperada más adelante y que se pueda buscar velozmente un dato entre miles o millones de otros, como hacen, por ejemplo, los buscadores de internet.
- **Las arquitecturas de computadoras.** Se refiere a los componentes que definen los distintos tipos de computadoras y cómo estos componentes se construyen a partir de la combinación de manipulaciones sencillas de voltaje eléctrico.
- **Las redes de computadoras.** Consiste en la forma en que las computadoras intercambian información permitiendo el funcionamiento de internet y de todas las aplicaciones que funcionan gracias a internet, como la web, la mensajería instantánea, los juegos en línea, las transmisiones de audio y video, etc.
- **Teoría de la Computación.** Reúne los fundamentos teóricos que marcan las diferencias entre los distintos lenguajes con sus posibilidades e imposibilidades, ventajas y desventajas, así como también otras áreas más específicas entre las que se encuentran ciertas áreas de la Matemática discreta, la Teoría de la complejidad y otras como la Computabilidad, que estudia qué problemas son efectivamente computables y cuáles no.
- **La [inteligencia artificial](#).** Se ocupa de la combinación de varias de las áreas previamente mencionadas para abordar problemas muy complejos mediante mecanismos que tienen puntos en común con la cognición humana. Incluye temas como aprendizaje automático, síntesis de información, reconocimiento de voz y de imágenes, etc.

Esta perspectiva sobre las Ciencias de la Computación tiene derivaciones a la hora de concebir una formación que permita el desarrollo de **una mirada crítica sobre la tecnología computacional**, su adopción, sus posibilidades y consecuencias, en contraposición a formaciones más operativas centradas en la capacidad de utilizarla de manera acrítica o incluso desconociendo aspectos centrales que tienen profundo impacto en la constitución del mundo contemporáneo. Así entendida, se trata de una disciplina que resulta fundamental tanto para comprender un mundo cada vez más organizado y dependiente de la tecnología como para poder participar activa y soberanamente en debates actuales sobre la interacción entre las computadoras y la sociedad.

Una propuesta formativa que parta desde esta concepción de las CC supone un enfoque pedagógico que promueva la indagación y la toma de decisiones argumentadas para fundamentar prácticas y cursos de acción. Los saberes a desarrollar deberán permitir enfrentar las modificaciones continuas que se producen en el campo de la computación y brindar autonomía para discernir ventajas y desventajas de su aplicación.

Este enfoque busca formar ciudadanas y ciudadanos con conocimientos de computación.

Por qué la escuela debe ser el lugar privilegiado para socializar estos saberes

Desde la Convención por los Derechos Humanos de 1948, la educación básica es considerada un derecho humano universal y, por lo tanto, se la declara obligatoria y gratuita.⁴ Luego de la Segunda Guerra Mundial (1939-1945), el acceso al conocimiento fue entendido como clave para garantizar el pensamiento crítico y la libertad. Los Estados firmantes de la Convención comprendieron que, sin educación formal, las ciudadanas y los ciudadanos no pueden de manera autónoma comprender y transformar el mundo en el que viven y asegurarse el acceso a otros derechos humanos tales como la salud, la vivienda, la participación política, etc.

En las últimas dos décadas, las tecnologías computacionales se han convertido en el medio de producción por excelencia. Atraviesan la mayoría de las actividades cotidianas (telecomunicaciones, transporte, operaciones bancarias, compra-venta de bienes y servicios, actividades recreativas, etc.) y el desconocimiento de conceptos del campo de las CC limita la reflexión crítica y la toma de decisiones adecuadas sobre temas delicados que afectan la libertad, el bienestar y los derechos en general.

A la luz de una mirada humanista, la escuela es depositaria de la tradición, la memoria, la cultura construida y los procesos de producción de la misma; esto debe ser compatible con comprender a la ciudadanía en un presente y ofrecer al alumnado saberes que permitan navegar e intervenir en el presente y en el futuro. Este triple rol de la institución educativa —como custodia del pasado, recapituladora de diversas experiencias del presente y constitutiva para nuevas en el futuro— es lo que determina que las decisiones curriculares se encuentren siempre en tensión (Dussel, 2007: 23):

El desafío es estructurar una idea de cultura común que pueda ser transmitida y compartida, que tome nota de las injusticias y los privilegios del pasado y que, al mismo tiempo, proponga algunas otras inclusiones que no vengan de la mano de la cultura del mercado o del individuo de autodiseño.

En otras palabras, incluir las CC en las aulas plantea un dilema respecto de los conocimientos que hoy se imparten en relación con la tecnología y la definición que se tome al respecto impactará en la formación docente tanto inicial como continua.

⁴ Convención por los Derechos Humanos, artículo 26, inciso 1.º: “Toda persona tiene derecho a la educación. La educación debe ser gratuita, al menos en lo concerniente a la instrucción elemental y fundamental. La instrucción elemental será obligatoria. La instrucción técnica y profesional habrá de ser generalizada; el acceso a los estudios superiores será igual para todos, en función de los méritos respectivos”.

En los últimos años, el concepto de alfabetización digital ha evolucionado para diferenciar entre usar la tecnología y consumir programas o aplicaciones de forma instrumental, y participar del mundo tecnológico a través de la comprensión de la computación, resolviendo problemas y creando nueva tecnología. Resulta interesante observar que siempre se habló de “alfabetización digital”, sin distinguir —como proponemos aquí— entre representación de la información (tecnología digital) y su procesamiento (tecnología computacional). Esto produjo variaciones en el énfasis de la formación, con predominancia de lo computacional en algunos momentos y de lo digital en otros (Martínez y otros, 2022).

Si bien la alfabetización digital requiere de saber usar los dispositivos y artefactos computacionales, limitarnos a ello en el contexto del sistema educativo supondría negar al estudiantado la posibilidad de comprender los modos de producción, funcionamiento y potencialidad de dichos dispositivos y artefactos que atraviesan nuestras prácticas y consumos. Esto afectaría principalmente el desarrollo económico, las oportunidades de participación y el liderazgo en posiciones de decisión de mujeres y diversidades.

Según datos de la Unesco (2019b), las mujeres y las niñas tienen un 25 % menos de probabilidades que los hombres de saber aprovechar la tecnología digital para fines básicos, cuatro veces menos de saber programar computadoras y 13 veces menos de solicitar una patente tecnológica. En 2010, Argentina reportó solo un 18 % de mujeres en el estudiantado de informática (Szenkman y Lotito, 2020). Otro informe de la Unesco (2019a) muestra cómo la desigualdad en el acceso al equipamiento, la conectividad y los saberes aumentan la brecha digital y cómo esta inequidad agrava la brecha de género: al tener menos oportunidades de estar en contacto frecuente y sistemático con estos conocimientos, las mujeres tienen menos oportunidades de interesarse por estos temas a lo largo de su vida, participar de los espacios de formación profesional y ejercer como tales en dicho ámbito. En relación con las diversidades ni siquiera hay datos.

La desnaturalización de patrones de comportamiento arraigados que reproducen desigualdades requiere de una acción positiva para que la ciudadanía en su conjunto tome conciencia y los revierta. Nuestra historia reciente está plagada de ejemplos del rol clave que cumple la escuela en esta tarea: la conciencia ecológica; los derechos de niños, niñas y adolescentes; la educación sexual integral, por mencionar algunos, se han convertido en temas del currículo en las últimas décadas. La tecnología computacional no escapa a esta necesidad.⁵ Su inclusión requiere revisar nuestras propias valoraciones y creencias, reflexionar sobre las expectativas de desempeño, dar cuenta de los múltiples modos en que la existencia de intereses, prejuicios y

⁵ Un estudio realizado en escuelas primarias de Córdoba (Martínez *et al.*, 2015, citado en Echeveste *et al.*, 2021) muestra cómo se desenvuelve el estudiantado por género frente a un mismo examen relacionado con conocimientos de programación. La variable que se observó en el estudio fue la formación vinculada a las CC recibida por las y los estudiantes. El estudio arrojó resultados contundentes: en aquellas escuelas en donde no se recibió formación específica en CC, los varones obtuvieron resultados significativamente mejores que las mujeres. En las escuelas donde estos saberes formaban parte del currículo escolar, la diferencia entre varones y mujeres fue imperceptible.

estereotipos impregnan los desarrollos tecnológicos y sus productos y visibilizar los aspectos históricos y sociales de las mujeres y las personas del colectivo LGTBIQ+ en las CC.

En definitiva, la alfabetización digital que las y los estudiantes necesitan para convertirse en ciudadanas y ciudadanos plenos, creadores de nuevas soluciones y herramientas y participantes informados en los debates de actualidad, incluye un conjunto de saberes bien definidos de las CC y un modo específico de abordaje.

El abordaje de las Ciencias de la Computación en el currículo argentino

En las últimas décadas, el Consejo Federal de Educación (CFE)⁶ ha construido acuerdos para definir los Núcleos de Aprendizaje Prioritarios (NAP) sobre los saberes provenientes de las Ciencias de la Computación para todo el territorio nacional.⁷

- En el 2006, el Ministerio de Educación de la Nación fijó lineamientos para las escuelas para la inclusión de contenidos y estrategias para la alfabetización digital y la enseñanza de la Programación. En la Ley de Educación Nacional 26.206, se incluyó la enseñanza de la tecnología: “[la necesidad de] desarrollar las competencias necesarias para el manejo de los nuevos lenguajes producidos por las tecnologías de la información y la comunicación [...] y promover el aprendizaje de saberes científicos fundamentales para comprender y participar reflexivamente en la sociedad contemporánea” (artículo 11 del Capítulo II, “Finés y objetivos de la política educativa nacional”).
- En el 2011, se acordaron los NAP para la Educación Tecnológica⁸ y entre las tecnologías a estudiar se incluyeron los procesos y medios técnicos computacionales.

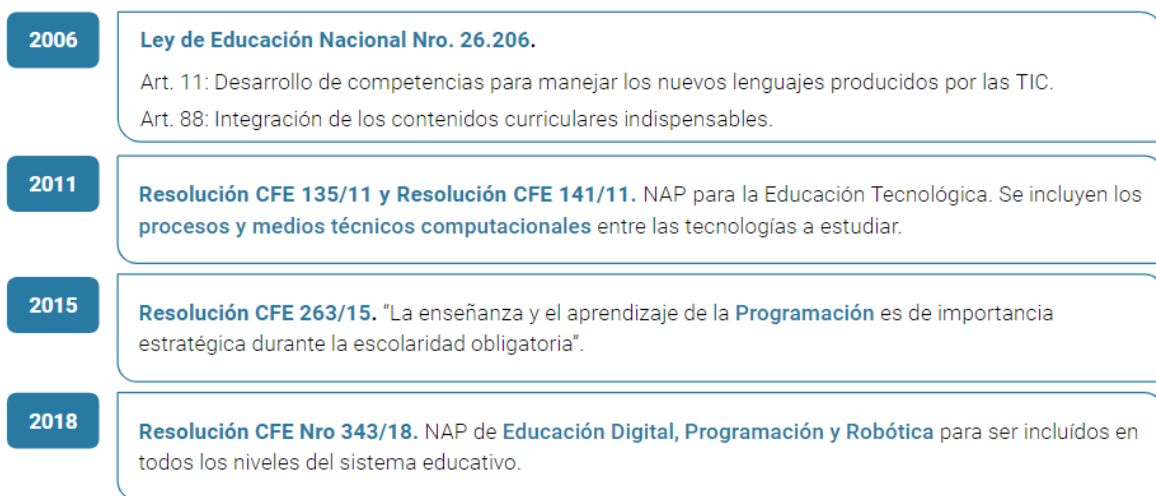
⁶ Organismo que decide y coordina la política educativa nacional con el objetivo de lograr un sistema educativo nacional integral y articulado. El CFE está formado por una Asamblea General, un Comité Ejecutivo y una Secretaría General. Está conformado por el ministro de Educación de la Nación, las máximas autoridades en materia educativa de las jurisdicciones del país y tres representantes del Consejo de Universidades.

⁷ En Argentina existen 25 ministerios (o Secretarías) de Educación (23 dependientes de las provincias, 1 de la Ciudad de Buenos Aires y 1 del Gobierno Nacional). De los primeros 24 dependen más de 46 000 escuelas públicas de gestión estatal que atienden a una población de 7,8 millones de niños, niñas y adolescentes, de los cuales asisten a nivel inicial 1,2 millones de estudiantes, a nivel primario 3,6 millones y al nivel secundario más de 3 millones. En cuanto a escuelas, alrededor de 16 000 son de nivel inicial, 19 000 de nivel primario y 11 000 de nivel secundario, aproximadamente. Funcionan 16 000 escuelas de gestión privada que en todos sus niveles atienden a cerca de 3 millones de alumnos y alumnas. En las escuelas públicas trabajan más de 550 000 docentes en todos los niveles (inicial, primario y secundario). En las escuelas privadas trabajan alrededor de 175 000 en todos los niveles. Del total de docentes: 40 000 son de materias afines a la Informática, aproximadamente. Datos obtenidos del Sistema de Consulta de Datos e Indicadores Educativos (SICDIE): <https://data.educacion.gob.ar/> y de la Dirección de Información Educativa del Ministerio de Educación de Argentina.

⁸ Ver Anexo III para el 2.º ciclo de la educación primaria, [Resolución CFE 135/11](#), y Anexo IV para el ciclo básico de la educación secundaria, [Resolución CFE 141/11](#).

- En el 2015, el CFE aprobó la resolución 263, que estableció que la Enseñanza de la Programación era de importancia estratégica.
- En el 2018 se aprobaron los NAP de Educación Digital, Programación y Robótica para **todos** los niveles del sistema educativo argentino.

Gráfico 1: Secuencia de reglamentaciones sobre educación en tecnología digital y computacional.



Fuente: Elaboración propia con datos del Ministerio de Educación de la Nación.

Ejemplos de implementación en algunas jurisdicciones

En el contexto de implementación de los NAP mencionados, algunas jurisdicciones están dando pasos concretos y graduales. En el 2018, la provincia de Neuquén⁹ aprobó un diseño curricular para nivel secundario, incluyendo un área que engloba contenidos de CC y matemáticas para el primer ciclo.

La provincia de Córdoba puso en marcha un programa de escuelas orientadas al desarrollo de *software* emplazadas en contextos vulnerables (Escuelas ProA)¹⁰. Y recientemente publicó una actualización de los contenidos de Educación Tecnológica para nivel primario y secundario.¹¹

Las provincias de Tucumán y Chaco iniciaron una reforma del currículo de nivel secundario con una reformulación de los contenidos de educación tecnológica y la inclusión de proyectos y contenidos de CC de manera progresiva.¹²

⁹ Gobierno de la provincia de Neuquén, Consejo Provincial de Educación, [Resolución 1463/18](#).

¹⁰ Gobierno de Córdoba, [Escuelas ProA](#).

¹¹ Gobierno de Córdoba, [Actualización de Educación Tecnológica](#).

¹² Unicef, [Programa PLaNEA](#).

En el contexto de la nueva escuela secundaria, la Ciudad Autónoma de Buenos Aires (CABA) incluyó contenidos de CC en la materia “Tecnologías de la Información” en los años tercero y cuarto de la educación secundaria.¹³

La provincia de La Pampa¹⁴ trabaja en el rediseño del currículo respecto de los contenidos de Educación Tecnológica para reorientarlos hacia la enseñanza de las CC de modo transversal en los niveles de educación inicial y primaria. Además, está diseñando un espacio curricular específico para el secundario.

La Rioja, con apoyo del Consejo Federal de Inversiones, ha comenzado un camino de planificación de su reforma. Santa Fe con apoyo del Banco de Desarrollo de América Latina ha encarado un ambicioso proyecto de reformulación de contenidos para todos los niveles educativos y de actualización docente para todos los perfiles del sistema.

Problemas que complejizan la inclusión de las CC en la educación obligatoria

La enumeración anterior muestra avances, pero también da cuenta de que, en la mayoría de las jurisdicciones, no se ha logrado aún incluir significativamente los saberes de las CC en las aulas. Esto podría deberse a una combinación de factores:

- la indeterminación respecto de si estos conocimientos deben (o no) suplantar los ya definidos por la resolución CFE 135/11, que fija los NAP de Educación Tecnológica, independientemente de si estos han sido incluidos como contenidos transversales o en una materia en el nivel;
- la confusión epistemológica generada por la resolución CFE 343/18 en relación con qué contenidos son los que se engloban bajo la denominación “educación digital, programación y robótica”, su sentido y su jerarquía;
- la dificultad de definir con precisión cuáles son las prácticas adecuadas por nivel que las graduadas y los graduados debieran desarrollar;
- la indefinición de la relevancia en términos de tiempo que el abordaje de estos saberes debería tener en cada nivel educativo;
- la omisión de cuál debería ser la actualización de las y los docentes en ejercicio y la formación inicial de quienes sean responsables en las aulas de la socialización de estos saberes.

¹³ CABA, Ministerio de Educación, *Propuestas didácticas por espacios curriculares, “Tecnologías de la Información”*.

¹⁴ “[Maccione firmó convenio con la Fundación Sadosky](#)”, *La Arena*, 11/07/2022.

Esta propuesta curricular intenta facilitar la definición de currículos, asistiendo así a los organizadores de las diferentes jurisdicciones que tienen la responsabilidad de avanzar en esta transformación y necesitan un marco claro en el que apoyarse para definir las propuestas de enseñanza de cada nivel.

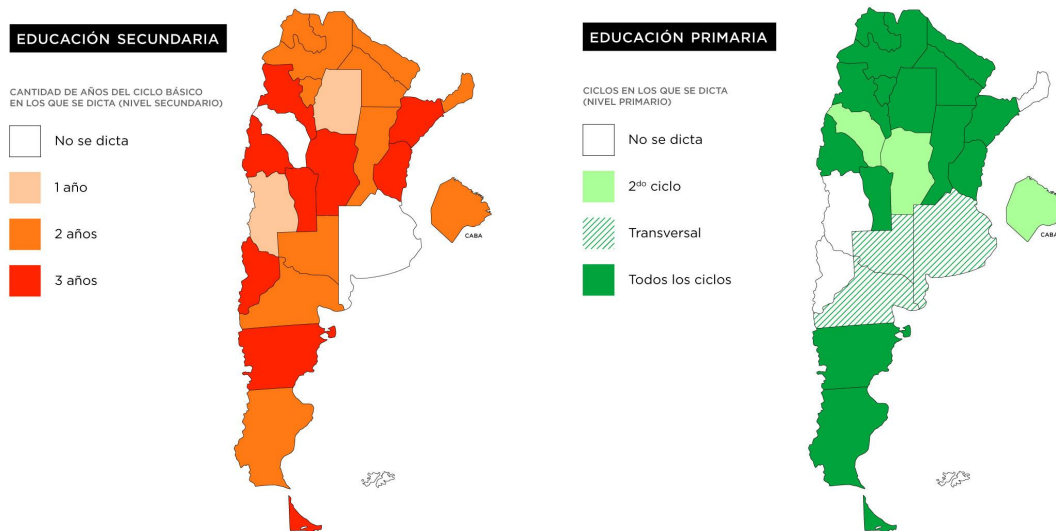
En nuestros intercambios con diferentes actores del escenario educativo, observamos que los temas que involucran el uso de TIC, la informática, la educación digital, la programación y la [robótica](#), el [pensamiento computacional](#), la [inteligencia artificial](#), lo digital, por citar algunos ejemplos, son utilizados con diferentes sentidos, en algunos casos como sinónimos. La falta de claridad conceptual sobre las Ciencias de la Computación como disciplina académica fue identificada por la Unesco (2019c) como uno de los desafíos clave para el desarrollo curricular. Por ello, se ha puesto especial énfasis en aportar claridad sobre el significado de términos computacionales usados con frecuencia en el ámbito educativo y explicitado en un [glosario](#) al final del documento.

Propósitos y enfoque de la propuesta curricular

Consideramos que el desarrollo de un currículo emancipador debe incluir saberes de las CC en el sistema educativo obligatorio y común para todo el alumnado, abordando con especial énfasis el rol de las tecnologías computacionales en la estructuración de vínculos sociales y la construcción de poder político y económico. La selección de contenidos de computación que construimos para las escuelas requiere de un conocimiento profundo de la disciplina para poder distinguir los saberes relevantes y centrales del área y definir estrategias didácticas que permitan el empoderamiento emancipador del conjunto social.

Con este propósito, se identificaron saberes específicos de las CC a ser enseñados por docentes con formación específica en la disciplina en un espacio curricular definido como obligatorio y de una frecuencia semanal. A partir de ello, para la educación inicial y el primer ciclo de la educación primaria, se recomienda centrarse en el uso de los [artefactos](#) y los [dispositivos computacionales](#). Para el segundo ciclo de la educación primaria y la educación secundaria, en esta propuesta curricular, se propone abordar las CC como objeto de estudio incorporando saberes sobre el funcionamiento de los artefactos y los dispositivos. Para la distribución de los saberes a desarrollar entre los ciclos y niveles se tuvo en cuenta la actual presencia de los contenidos de Educación Tecnológica en el aula y el hecho de que todas las jurisdicciones que han avanzado en la inclusión de contenidos de CC han optado por su actualización. Dado que la mayor parte de las jurisdicciones cuentan con un espacio curricular obligatorio en el primer ciclo de la secundaria destinado a la Educación Tecnológica, este es el nivel en donde se presentan la mayor parte de los saberes a abordar.

Gráfico 2. Presencia del espacio curricular de tecnología, educación tecnológica o informática en educación primaria y secundaria



Los propósitos generales de la propuesta son:

- *Desnaturalizar las [tecnologías digitales y computacionales](#)*: recuperar los conceptos que las explican y que fundamentan las decisiones sobre su adopción y las consecuencias prácticas y éticas de su implementación. La comprensión de estos conceptos permite concebir las tecnologías como un desarrollo humano en donde ni los artefactos ni los dispositivos son neutrales, sino que presentan ventajas, desventajas, similitudes y diferencias con otros contemporáneos y en el tiempo.
- *Focalizar la reflexión en los conceptos estables más allá del dispositivo del momento*: dado que las tecnologías digitales y computacionales se encuentran permanentemente en cambio, es relevante conocer los conceptos y los elementos estables que orientan el funcionamiento y el desarrollo de las tecnologías y que permiten enfrentar las modificaciones continuas que se producen en el campo (Zapata-Ros, 2018).
- *Valorar el impacto de las computadoras en la resolución de problemas*: centrar la enseñanza en los procesos de pensamiento para resolver problemas computacionales habilita reflexiones críticas sobre el potencial y las limitaciones de las tecnologías computacionales.
- *Relacionar los conceptos de computación con el funcionamiento de los dispositivos*: reconocer la relación de los conceptos con los dispositivos de uso cotidiano (de escritorio, celulares, tabletas, etc.) permite comprender profundamente el funcionamiento de estos últimos, su arquitectura, sus sistemas

operativos, sus aplicaciones y los programas y las formas en que pueden intercambiar información (redes e internet).

- *Comprender de qué modos la tecnología digital y computacional afecta los derechos individuales y colectivos en la vida contemporánea* y conocer estrategias y regulaciones que permiten protegerlos y/o ejercerlos.
- *Desarrollar los saberes necesarios para crear de manera autónoma artefactos computacionales* de su interés a partir de la participación en experiencias significativas y situadas de proyectos de programación.

Criterios para la selección de los contenidos

Para definir el recorte de los contenidos y las prácticas que permiten alcanzar los propósitos planteados se tuvieron en cuenta los siguientes criterios de selección de contenidos, tradicionalmente utilizados en las definiciones curriculares:

- *Seleccionar los conceptos de mayor nivel de generalidad.* Se han reconocido los conceptos generales que trascienden a los [dispositivos](#) y los [artefactos computacionales](#) y a problemas computacionales específicos. Al enseñar computación es fácil perderse en los detalles técnicos y operativos en perjuicio de la visión general de la disciplina (Bell, Tymann y Yehudai, 2018). Por ejemplo: a) para explicar qué es una computadora, en vez de enseñar la variedad de dispositivos computacionales que existen hoy en día con sus características particulares, se debería enseñar a reconocer en ellos un modelo de máquina programable al que responden todos los dispositivos y profundizar en ese modelo como objetivo del aprendizaje; b) para dar cuenta de la idea de programa, en lugar de enseñar con un único ejemplo específico (un lenguaje en particular aplicado a un problema en particular) o con unos pocos, se podría empezar por plantear un desafío que debe ser resuelto teniendo en cuenta una serie de potencialidades y limitaciones vinculadas a la lógica de funcionamiento.
- *Identificar conceptos fundamentales que son indispensables para comprender otros.* A partir de los aportes de curricularistas y especialistas en el contenido, autores de diferentes países han ofrecido la noción de “grandes ideas” para dar cuenta de los conceptos centrales de la disciplina (Bell, Tymann y Yehudai, 2018; Denning, 2004; Goode y Chapman, 2011; Simari, 2013). Por ejemplo, las estrategias de recuperación eficiente de información explican el funcionamiento de las bases de datos, pero también el mecanismo para enrutar mensajes en internet; las nociones de modularización están presentes en la programación con bibliotecas, pero también en la organización de los circuitos que

componen los dispositivos de hardware, en el diseño de redes y en la organización de sistemas operativos.

- *Elegir conceptos que promueven justicia.* Un criterio para la selección de contenidos podría ser la “justicia curricular” (Connell, 1997). Desde esta premisa y para el caso de las CC, la escuela debe ofrecer saberes que permitan atender las necesidades de conocimiento de sectores que no acceden a saberes digitales y computacionales por fuera de la escuela, para democratizar la participación ciudadana en el mundo computacional y achicar —con intención de cerrar— la brecha digital y, especialmente, la computacional. En este sentido, se abordan, por ejemplo, las normas vigentes que protegen a las infancias de delitos informáticos o acciones abusivas a través del uso de Internet y la necesidad de otras, aún faltantes, que amplíen el derecho de las persona a apropiarse de la tecnología computacional en su beneficio (por ejemplo, regulaciones relacionadas con la obsolescencia programada y el derecho a reparar).
- *Asumir una perspectiva crítica poniendo el foco sobre los aspectos problemáticos* de la existencia, el uso y la producción de tecnología digital y computacional en la actualidad, entendiendo que sus aspectos positivos son los más difundidos. Este enfoque busca problematizar los efectos que tienen o pueden tener los modos de producción y distribución de la tecnología sobre las desigualdades sociales vigentes y qué estrategias pueden las ciudadanas y los ciudadanos desplegar para mitigar los impactos negativos y potenciar sus usos emancipatorios.

Segunda parte. Propuesta curricular

En esta sección se organizan los saberes de las Ciencias de la Computación (CC) en cuatro áreas. Cada una de estas se organiza a su vez en ejes temáticos que agrupan conceptos o temas clave que se consideran relevantes para la enseñanza de esta disciplina en la escuela obligatoria.

La propuesta contempla el abordaje de las CC desde el segundo ciclo de la escolaridad primaria hasta el último ciclo de la secundaria. En ese marco se establecen los saberes a promover durante cada uno de los ciclos y el alcance y profundidad con que se espera abordar cada tema, teniendo en cuenta los saberes, experiencias y prácticas previas. La referencia a un ciclo escolar responde no solo a un abordaje que contempla una etapa evolutiva con intereses y posibilidades cognitivas, sino también obedece a una construcción espiralada de saberes, desde los conceptos más cercanos y concretos hasta los más generales y abstractos. Es así que algunos temas finalizan su recorrido luego de su abordaje en dos ciclos, mientras que otros, por su complejidad, se inician recién en la secundaria básica o incluso solo se abordan en el último ciclo de la secundaria. En la mayoría de los casos, los alcances esperados para un ciclo suponen haber incorporado los saberes del ciclo anterior. Dado que la inclusión de estos contenidos en las aulas es incipiente y seguramente su implementación sea gradual, las definiciones deberán guiarse por el hilo conductor de saberes y alcances más que por la pertenencia a un ciclo escolar.

El orden de aparición de las áreas y ejes no supone una jerarquía de ningún tipo ni tampoco una organización en el tiempo de la secuencialidad con la que deberían ser presentadas las propuestas de enseñanza para desarrollar los saberes esperados. En ese sentido, en el contexto de las situaciones de enseñanza que cada docente diseñe, se espera que los temas englobados por las distintas áreas se aborden de manera relacionada a propósito de una situación problemática o desafío a resolver.

Además, se distinguen cuatro [prácticas computacionales](#), que representan el “hacer computación”. Es en el marco de estas prácticas que los conceptos se construyen y adquieren sentido, por ello, resultan imprescindibles para poder “pensar” la computación. En este sentido, se espera que todas las propuestas de enseñanza abonen al desarrollo de estas prácticas computacionales.

Gráfico 3. Organización de la propuesta curricular.

Área →	ÁREA A		
Eje →	Eje 1		
Saberes →	Primaria, segundo ciclo	Secundaria, ciclo Básico	Secundaria, ciclo Orientado
	Saber 1 Saber 2	Saber 1 Saber 2	Saber 1
Temas ↓	Alcances ↓		
Tema 1	Alcance 1 Alcance 2	Alcance 1 Alcance 2	Alcance 1
Tema 2	Alcance 1	Alcance 1 Alcance 2	Alcance 1 Alcance 2
Prácticas →	Prácticas computacionales		

Gráfico 4. Síntesis de la propuesta curricular

ÁREAS	CIUDADANÍA Y COMPUTACIÓN	PROGRAMACIÓN	INFRAESTRUCTURA TECNOLÓGICA	DATOS E INTELIGENCIA ARTIFICIAL
EJES	Identidad digital	Soluciones a problemas computacionales	Organización y arquitectura de computadoras	Recolección y modelado de datos
	Estrategias para un uso seguro de internet	Representación de información en la resolución de problemas computacionales	Sistemas operativos	Ciencia de datos, inteligencia artificial y aprendizaje automático
	Computación y sociedad	Lenguajes de programación	Redes e internet	
PRÁCTICAS	Prácticas computacionales			

Áreas, ejes y contenidos

Teniendo en cuenta los diferentes saberes que involucran las CC y el recorte necesario para lograr los propósitos mencionados, en el apartado [Propósitos y enfoques de la propuesta curricular](#) se organizan los contenidos en las siguientes áreas y ejes temáticos:

A. ÁREA CIUDADANÍA Y COMPUTACIÓN

Eje A1 - Identidad digital

A1.1 - Convivencia en internet. Identidades digitales.

A1.2 - Información pública y privada compartida en internet.

A1.3 - Huella digital.

A1.4 - Vulneración de derechos y delitos informáticos. Grooming.

Eje A2 - Estrategias para un uso seguro de internet.

A2.1 - Estrategias para la protección de la información privada y la identidad digital: claves seguras, permisos y sitios seguros.

A2.2 - Estrategias para la protección contra software malicioso, rastreadores, sitios falsos y phishing.

Eje A3 - Computación y sociedad.

A3.1 - Desarrollos computacionales: evolución, oportunidades y brechas, intereses y necesidades. Su influencia en la vida y los derechos de las personas. Soberanía tecnológica.

A3.2 - Sesgos, prejuicios y estereotipos. Filtros en el acceso a la información en internet.

A3.3 - El problema del poder, el control y la vigilancia en las redes. Neutralidad de la red.

A3.4 - Desinformación a través de internet.

A3.5 - Modelos de producción y distribución de hardware y software.

A3.6 - Propiedad de los datos. Regulación, términos y condiciones.

A3.7 - Computación y ambiente: obsolescencia programada y derecho a reparar, extractivismo, demanda energética e impacto ambiental.

B. ÁREA PROGRAMACIÓN

Eje B1 - Soluciones a problemas computacionales.

B1.1 - Diseño de soluciones computacionales: estrategias de solución, modularidad y legibilidad.

B1.2 - Algoritmos y estructura de datos: problemas clásicos, complejidad y recursión.

B1.3 - Teoría de la computación: factibilidad y computabilidad.

Eje B2 - Representación de información en la resolución de problemas computacionales.

B2.1 - Nociones de representación de información.

B2.2 - Representaciones estandarizadas (imágenes, sonido, textos y números).

B2.3 - Modelado: representación con datos estructurados.

Eje B3 - Lenguajes de programación.

B3.1 - La sintaxis como un aspecto de los lenguajes de programación en bloques y texto.

B3.2 - La semántica como el significado de los programas y sus partes en términos del problema que resuelve.

B3.3 - Herramientas de lenguaje de programación.

B3.4 - Ejecución secuencial y paralela de programas.

B3.5 - Compilación e interpretación: intermediación entre el código fuente y la ejecución.

C. ÁREA INFRAESTRUCTURA TECNOLÓGICA

Eje C1 - Organización y arquitectura de computadoras.

C1.1 - La computadora como un sistema integrado por hardware y software.

C1.2 - Modelo de máquina programable: arquitectura von Neumann.

C1.3 - Componentes de hardware.

C1.4 - Ejecución a bajo nivel: lenguaje de máquina, representación binaria e impulsos eléctricos.

Eje C2 - Sistemas operativos.

C2.1 - El sistema operativo como administrador de recursos e interfaz para el hardware.

C2.2 - Abstracciones del sistema Operativo: archivos y sistemas de archivos, procesos, virtualización, usuarios.

Eje C3 - Redes e Internet.

C3.1 - Estructura y funcionamiento de redes de computadoras. Internet como una red.

C3.2 - Aplicaciones en internet: almacenamiento remoto, www, modelo cliente-servidor y P2P.

C3.3 - Mecanismos de seguridad en redes: protocolos, criptografía y anonimización.

D. ÁREA DATOS E INTELIGENCIA ARTIFICIAL

Eje D1 - Recolección y modelado de datos.

D1.1 - Recolección de datos activa y pasiva.

D1.2 - Modelado de datos: entidades con atributos.

Eje D2 - Ciencia de datos, inteligencia artificial y aprendizaje automático.

D2.1 - Aplicaciones de inteligencia artificial. Mecanismos generales.

D2.2 - Preparación, análisis y visualización de datos.

D2.3 - Modelos de aprendizaje automático.

D2.4 - Subjetividad expresada en los modelos. Sesgo algorítmico.

A. Área Ciudadanía y computación

Con la computación, la humanidad ha transformado las formas de producción de bienes y el tipo de productos existentes, la provisión de servicios y su variedad, las modalidades de esparcimiento, la vinculación, la formación y la comunicación de las personas y, por lo tanto, el modo en el que se construye y se participa de la vida contemporánea a nivel local, nacional y mundial.

La subjetividad, la organización política y las prácticas culturales son transformadas y reguladas por lo computacional. Estas transformaciones requieren ser analizadas como un producto humano no neutral, ya que responden y crean intereses, necesidades y relaciones de poder en un contexto histórico social específico. En este contexto, para alcanzar el ejercicio de una ciudadanía plena se requiere comprender las funcionalidades, las potencialidades y las limitaciones de la tecnología computacional. Sin ello, no es posible hacer uso responsable y seguro de las herramientas disponibles, participar de los debates éticos que estos cambios suponen, ni intervenir en la construcción del futuro de manera crítica.

Eje A1. Identidad digital

Los espacios digitales en internet como espacios de convivencia constituyen entornos donde ejercer una ciudadanía plena y activa, con derechos y responsabilidades. En la Argentina existe un marco legal que promueve, protege y regula algunos derechos digitales (por ejemplo, el acceso, la protección de datos, la integridad sexual, la libertad y la propiedad). La actividad que realizan las personas en este espacio de convivencia, que tiene características particulares, construye una identidad digital de cada una y deja un rastro, a veces voluntario y otras involuntario.

El conocimiento de esas particularidades del entorno digital, de los derechos de las personas, de pautas de autocuidado, de modos de proteger la privacidad de cada uno, de las formas que adquiere la violencia digital y la vulneración de derechos, resulta un pilar fundamental para convivir en entornos respetuosos y formar ciudadanas y ciudadanos comprometidos y responsables.

En la siguiente tabla se describen los alcances previstos para cada uno de los ciclos escolares en relación a los temas que pertenecen a este eje.

Área →	Ciudadanía y Computación		
Eje →	Identidad digital		
Saberes →	Primaria - 2º Ciclo	Secundaria - Ciclo Básico	Secundaria - Ciclo Orientado
	<p>El reconocimiento del espacio público de convivencia en internet en el que ciudadanas y ciudadanos participan a través de una identidad digital, y en el cual tienen derechos específicos y la responsabilidad de intervenir con respeto y sin violencia, conociendo estrategias para detenerla y comprendiendo que la actividad en internet deja una huella.</p> <p>El conocimiento de pautas de autocuidado, de la existencia de situaciones de vulneración de derechos en internet y de a quién recurrir frente a situaciones de violencia digital.</p>	<p>El ejercicio de una ciudadanía activa desde una identidad digital responsable, la participación en prácticas sociales propias de los entornos digitales y la mirada crítica sobre formas de violencia digital (discursos de odio, violencia de género, ciberbullying).</p> <p>El reconocimiento de las normas (resoluciones, leyes, ordenanzas, etc.) que protegen derechos digitales, de los modos en que cada persona puede gestionar la privacidad de sus datos personales y sensibles, de las modalidades que adoptan la vulneración de derechos, los delitos contra la integridad sexual (grooming) y otros delitos informáticos.</p>	<p>El ejercicio de una ciudadanía activa desde una identidad digital responsable, comprometida en la defensa de los derechos digitales y de los espacios de convivencia libres de violencia.</p>
Temas ↓	Alcances ↓		
	Primaria - 2º Ciclo	Secundaria - Ciclo Básico	Secundaria - Ciclo Orientado
1. Convivencia en internet. Identidades digitales.	Reconocer los entornos digitales como espacios de convivencia entre personas en los que pueden participar y deben interactuar con respeto; y que como	Comprender que los entornos digitales son espacios de convivencia donde se ejerce la ciudadanía , reconociendo derechos y responsabilidades propios y respetando y	Ser partícipes activos de espacios de convivencia en internet respetuosos.

ciudadanas y ciudadanos digitales se tienen **derechos y responsabilidades** desde el comienzo del uso de internet.

- Todas las personas tienen derecho a buscar, recibir y difundir información, siempre que no resulte perjudicial u ofensiva.
- La privacidad debe ser protegida y corresponde a cada persona decidir qué información propia puede ser visible.
- Se debe pedir permiso antes de registrar imágenes de otras personas o de subirlas a alguna plataforma o red social.

Reconocer la **existencia de la identidad digital** como una construcción creada a partir del conjunto de informaciones que cada persona publica en internet y redes sociales, y que cada quien puede decidir qué reflejar en los entornos digitales.

Reconocer que las personas contribuyen a construir la **identidad digital de otra** a partir de la información que sobre ella publican.

reconociendo los derechos y responsabilidades de las otras personas.

Conocer que en la Argentina existe una ley (Ley 27.078 Tecnologías de la información y las comunicaciones) que establece derechos digitales sobre el acceso a la tecnología.

- Todas las personas tienen derecho a usar computadoras, celulares y dispositivos electrónicos.
- Tienen derecho a tener internet sin importar edad o género, niveles de ingreso, ubicación geográfica o si tienen alguna discapacidad.

Ampliar el **horizonte cultural** reconociendo los espacios digitales como ámbitos de participación y compromiso donde ejercer una ciudadanía activa.

Vincular las nociones de **estructura y funcionamiento** de la red internet con las posibilidades de circulación de información, el alcance y la inmediatez para comprender los diferentes **usos y prácticas sociales** que esta estructura habilita.

Comprender la importancia de contar con y defender los **derechos digitales de la ciudadanía**.

Reconocer que la identidad digital forma parte de la vida cotidiana y de la identidad de las personas.

Reconocer que **es posible falsificar y/o robar la identidad digital** y a partir de ello se pueden generar situaciones de vulneración de derechos (por ejemplo, situaciones de grooming).

Reconocer que la convivencia en entornos digitales requiere, al igual que en entornos no digitales, el **respeto** de las personas, **sin discriminar, dañar ni perjudicar a otros** a través de la publicación de información en internet (por ejemplo, no ejercer violencia digital a través de la divulgación de contenidos de índole sexual sin consentimiento).

Reconocer al **ciberbullying** como una situación de **violencia digital entre pares**, a través de la utilización de medios de comunicación digitales para acosar u hostigar de forma sistemática y sostenida en el tiempo a una persona, mediante ataques personales, amenazas,

Identificar los **discursos de odio** que son favorecidos por algunas de las características de internet (como el anonimato o la falta de regulación y mediación de la información), para no apoyarlos ni promoverlos, así como también conocer las formas de denunciarlos¹⁵.

Analizar críticamente las **implicancias en la identidad digital propia y ajena** de la información que se publica (datos personales, imágenes, videos, noticias, comentarios, gustos, amistades), dónde se publica y con qué grado de visibilidad, teniendo en cuenta que los entornos digitales son espacios de convivencia entre personas. Así como reconocer la importancia de pedir el consentimiento si la información involucra a otra persona.

Recuperar los conocimientos sobre **ciberbullying**, y vincularlos con el desarrollo de una identidad digital responsable sin perjudicar a terceros.

¹⁵ Se entiende por “discursos de odio” ciertas narrativas sociales que circulan y se reproducen principalmente en el espacio público, ya sea en los medios de comunicación, redes sociales e internet, y que transmiten prejuicios y estereotipos sobre un grupo o colectivo de personas en particular, teniendo por objetivo justificar, legitimar e incitar la confrontación y/o la violencia social de un sector de nuestra sociedad sobre otro. (INADI, s/f [Una aproximación a los discursos de odio: antecedentes de investigación y debates teóricos. Coordinación de Investigaciones y Observatorios sobre Discriminación](#))

	<p>divulgación de información confidencial y/o falsa, entre otras estrategias.</p> <hr/> <p>Reconocer las particularidades del entorno digital (el anonimato, la ausencia física que permite no ver el daño causado, el alcance, la diversidad de formatos y plataformas, la dificultad de eliminar un contenido, la asincronicidad, entre otras) que alimentan la violencia, y no minimizar su impacto, ya que lo que ocurre en internet es parte de la vida de las personas.</p> <hr/> <p>Conocer estrategias para detener el ciberbullying, asumir la responsabilidad individual de no circular ni reforzar acciones de hostigamiento y conocer formas de denunciar la violencia tanto en plataformas digitales como recurriendo a un adulto de confianza.</p>		
<p>2. Información pública y privada compartida en internet.</p>	<p>Reconocer ámbitos digitales públicos y privados para formar criterios sobre qué</p>	<p>Conceptualizar qué son los datos personales (información de cualquier tipo asociada a la identidad de una persona: nombre completo, documento, datos de localización, teléfono, la</p>	

datos y con quién es seguro compartir información.

Reconocer la necesidad de cuidar la propia **privacidad** y respetar la privacidad de otras personas (por ejemplo, pedir permiso a la hora de publicar imágenes o videos de otras personas en redes sociales o grupos de mensajería) empleando estrategias para la protección de la información privada y la identidad digital.

imagen personal, etc.¹⁶).

Comprender que los datos personales son **privados**, por qué es importante cuidarlos y saber que debe mediar el **consentimiento** de la persona a la que pertenecen para que otras personas u organizaciones los utilicen.

Conocer que existe una **ley de protección de datos personales**¹⁷ (Ley 25.326) que protege a las ciudadanas y ciudadanos si sus datos de identidad, salud o crédito son usados sin su consentimiento, y que garantiza que puedan saber quién, para qué, cuándo y por qué se utilizan sus datos personales guardados en archivos, registros, bancos de datos públicos o privados.

Saber que nadie puede obligarnos a compartir **datos sensibles** que revelen origen racial y étnico, opiniones políticas, convicciones religiosas, filosóficas o morales, afiliación sindical e información referente a la salud o a la vida sexual, y que es una **decisión personal** hacerlos públicos en función de cómo cada quien quiera construir su identidad digital.

Comprender por qué los datos que puedan ser usados para identificar, contactar o localizar a

¹⁶ <https://www.argentina.gob.ar/justicia/convosenlaweb/situaciones/como-protoger-mis-datos-personales>

¹⁷ <https://www.argentina.gob.ar/justicia/derechofacil/leysimple/datos-personales>

		<p>una persona, datos bancarios y de crédito son también datos sensibles, y formar criterios para determinar cuándo las acciones personales exponen en forma pública estos datos propios o de terceros pudiendo comprometer la seguridad o privacidad de una persona.</p> <hr/> <p>Gestionar la privacidad identificando ventajas, desventajas, riesgos y posibilidades al compartir información como pública y como privada en entornos digitales para decidir qué y cómo publicarla.</p>	
<p>3. Huella Digital</p>	<p>Reconocer que al navegar en internet se deja un rastro de información, compuesto tanto por información que generamos de manera voluntaria (poner “me gusta”, escribir un comentario a una publicación, etc.) como por el registro automático y de manera continua de nuestras acciones en internet (clics, sitios visitados, búsquedas, tiempo de permanencia, ubicación geográfica, etc.).</p>	<p>Conceptualizar el fenómeno de la huella digital como todos los datos que resultan de nuestra actividad en internet, teniendo en cuenta que los datos que generamos para que sean registrados (un “me gusta”, comentarios, suscripciones, etc.) y los que resultan del registro automático y continuo de nuestras acciones en internet (clics, sitios visitados, búsquedas, tiempo de permanencia, ubicación geográfica, etc.) quedan registrados y eventualmente pueden ser utilizados.</p> <hr/>	<p>Reflexionar acerca de la huella digital como un rastro económicamente valioso.</p>

		<p>Formar criterio para decidir si se deja información de forma voluntaria, considerando que deja un rastro que eventualmente pueda ser usado para fines como publicidad dirigida, campañas políticas personalizadas, recomendaciones de contenido, predicciones, etc.</p>	
<p>4. Vulneración de derechos y delitos informáticos. Grooming.</p>	<p>Apropiarse del derecho a elegir quién puede ver el cuerpo propio o las imágenes donde aparezca en entornos virtuales y saber que ninguna persona puede exigir ni presionar a otra para que muestre su cuerpo, ni causarle incomodidad, y que estas situaciones, emprendidas por un adulto, se conocen como grooming y constituyen un delito de abuso a la integridad sexual.</p> <p>Reconocer que una persona que se conecta con nosotros a través de internet puede tener un perfil falso y simular ser otra persona o tener nuestra misma edad, para emprender una situación de grooming.</p> <hr/>	<p>Identificar las formas particulares que adquiere la violencia de género en entornos digitales (la reproducción de estereotipos; los discursos de odio; recibir contenidos sexuales explícitos sin consentimiento; exigir el envío de imágenes o información íntima; difusión de imágenes íntimas sin consentimiento; sextorsión; ejercer la vigilancia digital exigiendo compartir ubicación, compartir contraseñas, mostrar contactos o conversaciones, etc.) y conocer las formas de buscar ayuda.</p> <hr/> <p>Conceptualizar el grooming como toda acción por la que una persona adulta contacta a una niña, un niño o adolescente a través de entornos digitales para atentar contra su integridad sexual, saber que se encuentra</p>	

	<p>Comprender la importancia de recorrir a una persona adulta de confianza ante los primeros indicios de situaciones de grooming u otra situación de violencia digital o delito informático.</p>	<p>penado (Ley 27590¹⁸); adquirir habilidades para reconocer estas situaciones y saber qué hacer ante un caso de grooming, así como conocer sobre la existencia de canales estatales de ayuda.</p> <hr/> <p>Reconocer modalidades de delitos informáticos¹⁹ contra la libertad, la propiedad (estafas), entre otros, para cuidarse, buscar ayuda y vincularlo a las nociones de infraestructura y mecanismos de seguridad en redes (por ejemplo, tener los cuidados necesarios al usar una red wifi de acceso público).</p>	
--	---	--	--

Eje A2. Estrategias para un uso seguro de internet

Una de las necesidades originadas por el uso de tecnologías computacionales para el ejercicio de la ciudadanía reside en la adopción de medidas de seguridad frente al uso de sitios y aplicaciones en internet para prevenir delitos informáticos, robo de identidad en internet y otros daños. Tomar conciencia de los riesgos permite comprender la importancia de protegerse a través de claves seguras, permisos otorgados, y adquirir criterios para reconocer sitios inseguros, malware, sitios falsos y otras actividades sospechosas en la web.

En la siguiente tabla se describen los alcances previstos para cada uno de los ciclos escolares en relación a los temas que pertenecen a este eje.

¹⁸ <https://www.argentina.gob.ar/desarrollosocial/grooming>

¹⁹ <https://www.argentina.gob.ar/justicia/derechofacil/leysimple/delitos-informaticos>

Área →	Ciudadanía y Computación		
Eje →	Estrategias para un uso seguro de internet		
Saberes →	Primaria - 2º Ciclo	Secundaria - Ciclo Básico	Secundaria - Ciclo Orientado
	<p>La adopción de medidas de seguridad para el uso de tecnologías digitales, atendiendo a criterios de seguridad de la información, privacidad y cuestiones éticas vinculadas a la información.</p>	<p>La anticipación, descripción y desarrollo de medidas de seguridad informática para prevenir situaciones de acoso, delitos informáticos y robo de identidad en internet.</p>	<p>La anticipación, descripción y desarrollo colectivo de medidas de seguridad digital para usuarios, a través de la elaboración y apropiación de formas de proceder para prevenir y protegerse de ataques y acoso cibernéticos (por ejemplo, <i>phishing</i>, <i>malware</i> y <i>ransomware</i>, <i>ciberbullying</i>, etc.) y del uso responsable de información privada publicada en aplicaciones en Internet (tales como redes sociales, videojuegos, servicios de mensajería, etc.).</p>
Temas ↓	Alcances ↓		
1. Estrategias para la protección de la información privada y la identidad digital: claves seguras, permisos y sitios seguros.	Primaria - 2º Ciclo	Secundaria - Ciclo Básico	Secundaria - Ciclo Orientado
	<p>Reconocer la necesidad de hacer un uso seguro de la tecnología para resguardar su identidad digital de posibles robos, alteraciones o situaciones de violencia.</p> <hr/> <p>Reconocer que las imágenes, videos, noticias, comentarios, gustos que se</p>	<p>Afianzar la práctica de generación y utilización de claves de acceso (contraseñas) y los criterios de seguridad asociados a estas.</p> <hr/> <p>Diferenciar claves seguras y no seguras.</p>	<p>Apropiarse y generalizar estrategias para la protección de la información privada y la identidad digital.</p>

publican de forma activa en internet pueden tener distintos grados de visibilidad y las personas pueden decidir, en cierta medida, esos **grados de visibilidad**.

Comprender la noción de **claves de acceso** (contraseñas), la importancia de que estas sigan criterios de seguridad como forma de resguardar la identidad digital y evitar las consecuencias no deseadas (en particular, que cualquier persona que conozca la clave de acceso a una identidad digital pueda apropiarse de ella).

Conocer algunos **criterios de elección de claves** que resulten seguras (no palabras del diccionario, no secuencias de letras consecutivas, etc.) y las maneras de utilizarlas para no alterar esa seguridad (como no compartirlas con nadie, no anotarlas en lugares obvios o fácilmente ubicables por cualquiera, no usar la misma clave para diferentes sitios, etc.).

Reconocer cuándo un sitio utiliza un **protocolo seguro (https)** para proteger la información que intercambia con el usuario.

Reconocer **situaciones que involucran contraseñas y que suponen un riesgo para la seguridad** y saber cómo manejarlas (por ejemplo, no usar la contraseña en una red pública, no usar la misma contraseña para diferentes aplicaciones, modificar la clave de forma periódica).

Conocer los **tipos de permisos** disponibles en diferentes aplicaciones de internet, y cómo la administración de esos permisos permite controlar los **grados de visibilidad**, protegiendo la privacidad y la identidad digital.

Adquirir criterios para evaluar las **garantías de seguridad** (acceso con clave, autenticación, comunicación con protocolos seguros y solicitud solo de los permisos necesarios) que ofrece un sitio o aplicación en relación con la información personal que solicita (por ejemplo, proveer datos de tarjeta de crédito en sitios seguros, no brindar permisos de acceso a contactos a una aplicación de linterna).

Considerar otras estrategias para **proteger los datos personales** (usar el modo incógnito para que no se guarden las contraseñas y el historial de navegación, no usar la misma contraseña para los sitios a los que se accede y para las redes sociales, no ingresar datos personales en sitios desconocidos, no responder correos electrónicos donde solicitan que se completen

		<p>datos personales, leer los términos y condiciones de uso de los datos personales que proponen las redes sociales o aplicaciones antes de aceptarlas, no guardar contraseñas en lugares públicos, en sitios web que requieren el ingreso de usuario y contraseña, verificar que la dirección de la página sea auténtica²⁰).</p> <hr/> <p>Considerar los cuidados necesarios para sextear de forma segura²¹ sin que la información intercambiada se vuelva pública (el sexting forma parte de las maneras de expresar la sexualidad, en este caso particularmente a través de medios digitales).</p>	
<p>2. Estrategias para la protección contra software malicioso, rastreadores, sitios falsos y <i>phishing</i>.</p>	<p>Reconocer la existencia de sitios falsos tomando como indicio el nombre de la dirección web, formas de redacción del contenido y aspecto de las imágenes y certificaciones validadas (ssl).</p> <hr/> <p>Reconocer aplicaciones de celular confiables tomando como indicio los permisos que solicita, las reseñas de otros</p>	<p>Reconocer la existencia de mecanismos para recopilar información de navegación en internet (cookies, rastreadores), sitios o aplicaciones diseñadas con propósitos dañinos (software malicioso o malware) y complementos para el navegador que los identifican o bloquean.</p> <hr/> <p>Identificar mensajes fraudulentos que contienen enlaces donde se solicitan datos</p>	<p>Apropiarse y generalizar estrategias para la protección contra <i>malware</i>, sitios falsos y <i>phishing</i>.</p> <hr/> <p>Apropiarse y generalizar estrategias para la protección de la huella digital.</p>

²⁰ <https://www.argentina.gob.ar/justicia/convosenlaweb/situaciones/como-cuidar-los-datos-en-la-nube>

²¹ <https://www.argentina.gob.ar/justicia/convosenlaweb/situaciones/como-hago-para-sextear-seguro>

	<p>usuarios, la cantidad de descargas que tiene y el autor.</p> <hr/> <p>Atender a que las descargas de programas o aplicaciones de internet contengan solo lo necesario antes de ejecutar una instalación.</p>	<p>privados sensibles (por ejemplo, credenciales de acceso o números de tarjeta de crédito) o se induce a acciones perjudiciales (como el envío de dinero a cuentas ajenas), práctica que se conoce como phishing.</p> <hr/> <p>Comprender que el usuario es un participante activo de los ataques producidos por <i>malwares</i>, uso de sitios inseguros y <i>phishing</i>.</p>	
--	---	--	--

Eje A3. Computación y sociedad

Dadas las consecuencias y transformaciones que el desarrollo computacional introdujo para el ejercicio de la ciudadanía, se hace indispensable visibilizar el entramado en el que están insertos los artefactos computacionales a la luz de los intereses que motivan su producción y distribución, las brechas y desigualdades, la concentración de desarrollos computacionales, el poder y la soberanía o el impacto ambiental, en un contexto histórico social y regional específico.

El objetivo es que las y los estudiantes construyan una mirada más completa que se aleje de la visión simplista de considerar a la tecnología como un producto neutral y favorezca una actitud crítica, que promueva y exija los derechos digitales de la ciudadanía a partir de una computación con justicia social.

En la siguiente tabla se describen los alcances previstos para cada uno de los ciclos escolares en relación a los temas que pertenecen a este eje.

Área →	Ciudadanía y Computación
--------	--------------------------

Eje →	Computación y sociedad		
Saberes →	Primaria - 2º Ciclo	Secundaria - Ciclo Básico	Secundaria - Ciclo Orientado
	La construcción de una mirada no neutral sobre los artefactos y dispositivos computacionales, su desarrollo y distribución, así como la identificación de las consecuencias sociales y ambientales.	El reconocimiento de los intereses que guían el desarrollo de artefactos y dispositivos computacionales, las consecuencias y desigualdades para la ciudadanía.	La actitud crítica sobre los artefactos y dispositivos computacionales en términos de poder y concentración, las consecuencias para la ciudadanía y la soberanía, y la necesidad de contar con un marco regulatorio que proteja los derechos de la ciudadanía.
Temas ↓	Alcances ↓		
	Primaria - 2º Ciclo	Secundaria - Ciclo Básico	Secundaria - Ciclo Orientado
<p>1.</p> <p>Desarrollos computacional es: evolución, oportunidades y brechas, intereses y necesidades. Su influencia en la vida y los derechos de las personas. Soberanía tecnológica.</p>	<p>Identificar problemas resueltos computacionalmente en situaciones cotidianas.</p> <hr/> <p>Reconocer desde una perspectiva historiográfica la forma en que han evolucionado los artefactos y dispositivos computacionales de su interés y uso diario (cambios en la interacción, el problema que resuelven y la forma de resolverlo).</p> <hr/>	<p>Reconocer que los desarrollos computacionales influyen en las costumbres de las personas y en sus consumos culturales, crean nuevas necesidades, etc.</p> <hr/> <p>Reflexionar sobre el hecho de que el desarrollo de dispositivos y artefactos computacionales es motivado por intereses y necesidades de personas, organizaciones u empresas.</p> <hr/> <p>Reconocer la influencia desigual en los desarrollos computacionales de los intereses económicos o políticos de grupos</p>	<p>Reflexionar sobre cómo la irrupción de nuevos artefactos tecnológicos y computacionales influyeron e influyen en la modificación de actividades laborales y profesionales, y reflexionar sobre las consecuencias de ello en la vida de las personas.</p> <hr/> <p>Reflexionar acerca de las implicaciones sociales, políticas y económicas de la concentración de servicios y productos tecnológicos en pocas empresas (por ejemplo, con respecto a los derechos de las personas).</p>

Reconocer que el uso de dispositivos y artefactos computacionales **impacta en forma positiva o negativa** en la salud de las personas, en los modos de comunicarse, entretenerse o relacionarse, en la reproducción de estereotipos (por ejemplo, los juegos destinados a niños y los destinados a niñas) y brechas de género.

reducidos de personas, organizaciones u empresas por sobre los de la población.

Comprender desde una perspectiva histórica, política y de género el desarrollo de la computación.

Reconocer que la **desigualdad de género** adquiere nuevas formas en internet:

- las mujeres y personas del colectivo LGTBQ+ tienen menos posibilidades de acceder, saber y usar (brecha digital), tienen menos representación en la industria del software y, por tanto, los desarrollos no responden a intereses y necesidades diversos.
- Los contenidos digitales que circulan de forma masiva reproducen estereotipos de género.

Reconocer el desigual desarrollo computacional en diferentes partes del mundo reflexionando particularmente sobre el **desarrollo local en contraste con el mundial**, y las consecuencias de este contraste con el desarrollo de la **soberanía tecnológica** a nivel nacional.

Reflexionar sobre cómo una solución tecnológica o computacional generada en una región e impulsada de modo global puede **impactar de modo diferente en culturas o sociedades y generar nuevos problemas**.

Comprender la importancia de contar con un marco regulatorio y exigir que se aplique, para proteger los **derechos de las personas ante las consecuencias de desarrollos computacionales**.

Ser partícipes activos y agentes de cambio en la promoción de la equidad de género vinculada a los desarrollos computacionales.

2.

Sesgos, prejuicios y estereotipos. Filtros en el acceso a la información en internet.

Reconocer que los artefactos computacionales no son objetivos, sino que reflejan la **mirada de las personas y las organizaciones** que los desarrollan e implementan.

Reconocer la existencia de un **recorte en la información** a la que accedemos en internet (por ejemplo, en resultados de búsquedas, en las publicaciones que se ven en redes sociales, en las recomendaciones de plataformas, etc.).

Reconocer que existen **criterios e intereses** según los cuales los artefactos computacionales realizan el **recorte en la información** a la que accedemos en internet (personalizaciones, legislación, intereses de los propietarios), conocido como **burbuja de filtros**²², y que puede generar censura u homogeneización de la información.

Reconocer de qué manera los artefactos computacionales están atravesados por las **subjetividades e intereses** de quienes los desarrollan (por ejemplo, una herramienta de recolección que no contempla entre sus opciones diversas opciones de género, o soluciones computacionales que al requerir determinados dispositivos establecen nuevas brechas de acceso).

Reconocer situaciones en las que el sesgo algorítmico **reproduce o refuerza prejuicios o discriminaciones** y dimensionar sus consecuencias en términos humanos, como por ejemplo, la censura, el aislamiento, la amplificación de estereotipos, etc.

²² El concepto de burbuja de filtros se refiere a la selección personalizada de la información que recibe cada individuo al consumir contenido en internet. Esta lo hace sentir cómodo, pero lo aísla de las y los demás. Los algoritmos que utilizan las redes permiten recopilar todos los rastros de información digital que va dejando cada cibernauta —desde búsquedas en Google a “Me gusta” en Facebook—, y van aprendiendo sobre esa persona para ofrecerle una visión del mundo ajustada a sus preferencias. De esta forma, un mismo término introducido por dos personas distintas en un motor de búsqueda no arrojará ni la misma cantidad ni los mismos resultados para una y para otra, sino que los adaptará en función de lo que conoce de cada una, gustos personales, creencias, orientación ideológica. (Conferencia de Eli Pariser en Ciclo [#TechSociety, 2017](#), Fundación Telefónica).

		<p>Tomar conciencia de que la percepción que se construye de la realidad a partir de la información a la que accedemos en internet es parcial y sesgada.</p> <p>Comprender la necesidad de contar con un marco regulatorio y exigir que se aplique, para proteger los derechos de las personas ante las consecuencias del desarrollo y aplicación de los artefactos computacionales basados en datos (por ejemplo, para qué tipos de problemas se pueden aplicar modelos de inteligencia artificial).</p>	
<p>3. El problema del poder, el control y la vigilancia en las redes. Neutralidad de la red.</p>		<p>Identificar cómo la estructura de internet hace posible formas de vigilancia sobre el comportamiento de las personas (por ejemplo, a través del análisis del tráfico de datos) y cómo el control de tráfico favorece o limita la neutralidad de la red²³.</p>	<p>Conocer y adquirir conciencia sobre la responsabilidad que conlleva el poder que tienen los proveedores de servicios de internet, en tanto pueden favorecer o limitar la neutralidad de la red y ejercer formas de vigilancia y control.</p>

²³ “La **neutralidad de la red** fue concebida para proteger el carácter libre, abierto e innovador de internet, para lo cual se garantizan dos cosas: cualquiera debe poder conectarse y la información debe fluir lo más libremente posible [...] la neutralidad de la red establece que los operadores de redes (incluidos los proveedores de servicio de internet al hogar) no deben discriminar arbitrariamente entre la información que pasa por su infraestructura. Es decir, deben permitir todo uso legítimo de la red sin discriminar arbitrariamente por origen, destino, dispositivos, protocolos o puertos.” (Javier Pallero, 2018, “[Neutralidad de la red: ¿qué te puedo cobrar?](#)”, El Gato y la caja). La ley Argentina digital 27078 en sus artículos 56 y 57 contempla este derecho de la ciudadanía y regula a los proveedores de servicios de internet.

			<hr/> <p>Comprender la importancia de contar con un marco regulatorio y exigir que se aplique, para controlar la neutralidad de la red, la vigilancia y el control en función de los beneficios o derechos digitales de la ciudadanía.</p>
<p>4. Desinformación a través de internet.</p>	<p>Tomar conciencia de que no toda la información que circula en internet es verdadera.</p> <hr/> <p>Reconocer que hay información que circula más que otra y que por lo general la información falsa o engañosa tiene un alto nivel de circulación.</p> <hr/> <p>Reconocer la responsabilidad individual en la circulación de la información.</p>	<p>Reconocer cómo influye en la población la propagación de discursos de odio, la generación y circulación masivas de información falsa o engañosa.</p> <hr/> <p>Conocer mecanismos para la viralización de información falsa (por ejemplo, <i>bots</i> y <i>troll centers</i>) y cómo los algoritmos contribuyen a amplificarla.</p> <hr/> <p>Reconocer la existencia de soluciones computacionales que facilitan la generación y propagación de información falsa (<i>deep fakes</i> a partir de montajes de video o sintetizadores de voz) y el trabajo de análisis manual para</p>	<p>Reconocer las consecuencias de la circulación de la información falsa o engañosa, los discursos de odio, como herramientas de poder.</p> <hr/> <p>Reconocer los intereses y motivaciones de las personas o instituciones detrás de manipulación de la información (por ejemplo, noticias falsas con la intención de confundir, manipular las opiniones y alterar el curso de ciertos sucesos como los electorales).</p> <hr/> <p>Incorporar estrategias y mecanismos para identificar cuándo una noticia puede ser</p>

		<p>detectar si una información es falsa o engañosa.</p> <hr/> <p>Conocer e implementar estrategias para detectar información falsa (verificación de fuentes, distinguir si se refieren a hechos u opiniones).</p>	<p>falsa y saber cómo proceder en dicho caso para controlarla, denunciarla, etc.</p> <hr/> <p>Reconocer que la desinformación utiliza como vehículo los discursos de odio, que al apelar a la emocionalidad promueve mayores reacciones en las personas que ayudan a amplificar su difusión.</p> <hr/> <p>Comprender la importancia de contar con un marco regulatorio, y exigir que se implemente, para denunciar y penalizar la información falsa.</p>
<p>5. Modelos de producción y distribución de hardware y software.</p>	<p>Reconocer que existen diferentes modos de producir software y hardware: producción abierta y colaborativa, producción cerrada.</p> <hr/> <p>Reconocer que existen diferentes modos de distribuir el software y el hardware: licencias libres y licencias privativas, sus ventajas y desventajas en términos de nivel de dependencia y</p>	<p>Identificar al software como una producción intelectual (de personas, organizaciones u empresas) y, por lo tanto, sujeta a las nociones de derecho de autoría.</p> <hr/> <p>Reconocer que quienes desarrollan software disponen de instrumentos legales para habilitar o restringir modos de producción y de distribución (por ejemplo, las licencias Creative Commons vs. las patentes), y las</p>	<p>Vincular el desarrollo de instrumentos legales con la defensa de algunos de los intereses de productores o personas usuarias.</p> <hr/> <p>Valorar la producción colaborativa de software como una forma de garantizar las libertades (libertad de usar, de estudiar, distribuir y mejorar) y los derechos vinculados a los artefactos y dispositivos computacionales.</p>

	<p>posibilidad de análisis y apropiación para modificar y crear artefactos computacionales.</p> <hr/> <p>Reconocer cuáles de los artefactos o dispositivos computacionales que utilizamos a diario están basados en software y hardware libre, ya sea que hayan sido producidos de forma cerrada y con licencias privativas o de forma abierta y con licencia libre (por ejemplo, sistemas operativos comerciales basados en software libre).</p>	<p>características de estos.</p> <hr/> <p>Reconocer la relación entre las motivaciones e intereses de productores y usuarios de software y hardware con los modos de producirlos o distribuirlos.</p> <hr/> <p>Reconstruir desde una perspectiva historiográfica cómo se ha producido y compartido el software y el hardware desde los inicios hasta el desarrollo de los grandes monopolios.</p> <hr/> <p>Apropiarse de prácticas de participación computacional, cultura libre y comunicabilidad de los programas desarrollando y utilizando bibliotecas de funciones y procedimientos colaborativas con archivos y programas legibles y comprensibles por la comunidad de esa aula.</p>	
<p>6. Propiedad de los datos. Regulación, términos y condiciones.</p>	<p>Reconocer que el control del usuario sobre los datos que sube o comparte en una plataforma de internet está mediado por las empresas dueñas de los servicios que almacenan los datos</p>	<p>Identificar que la información y los datos que se suben o comparten en internet son propiedad de las empresas dueñas de los servicios y las posibilidades que esto les</p>	

(por ejemplo, cuando el usuario solicita borrar datos, es la empresa la que tiene el control de los datos físicos y decide si los borra o no, más allá de que el usuario los vea como borrados).

otorga (por ejemplo, la potestad de lucrar comercialmente con ellos o venderlos).

Reconocer que la propiedad y uso de datos a gran escala puede ser una herramienta para influir en fenómenos sociales (elecciones, campañas publicitarias, etc.).

Tomar conciencia de que al aceptar los **términos y condiciones** de una plataforma se obtienen servicios a cambio de algo (por ejemplo, que conozcan nuestros gustos, preferencias, nuestra interacción con el servicio y otra información).

Reflexionar sobre las consecuencias del uso que hacen las empresas u organismos públicos propietarios de los datos, vinculando con la necesidad de instrumentos de regulación (términos y condiciones de las plataformas, leyes de protección de datos personales, derecho al olvido, etc.).

7.

Computación y ambiente: obsolescencia programada y derecho a reparar, extractivismo, demanda energética e impacto ambiental.

Reconocer el impacto ambiental asociado a la **producción y descarte** de dispositivos computacionales (la basura tecnológica, extractivismo y etc.).

Reconocer que los dispositivos computacionales (celulares, computadoras, televisores inteligentes, etc.) se cambian en periodos de tiempo relativamente cortos por motivos de diseño por parte de los fabricantes (**obsolescencia programada**²⁴).

Reconocer el impacto ambiental de la **demanda de infraestructura física y energética** de los servicios de internet (como redes sociales, buscadores, servicios de mensajería, videojuegos en línea, etc.).

Comprender quiénes se benefician y quiénes se perjudican con la obsolescencia programada y el **derecho a reparar**.

Tomar conciencia de la dimensión de la demanda de infraestructura física y energética de los servicios de internet (como redes sociales, buscadores, servicios de mensajería, video, etc.) y su **impacto ambiental asociado a los grandes centros de cómputos**.

Poner en relación el impacto de la producción y uso de dispositivos y artefactos computacionales con la noción de **sustentabilidad** definida por la ley de Educación Ambiental Integral.²⁵

²⁴ La obsolescencia programada es la vida útil que le da una fábrica o empresa a un producto, cuando pase este periodo el producto se volverá inservible. La mayoría de los productos están "programados para morir". Muchas veces cuando estos dispositivos dejan de funcionar es más económico comprar uno nuevo que reparar el que ya tenemos (ya sea por el costo de los repuestos o porque ya salió un modelo más actualizado). La obsolescencia programada se creó para que el consumidor se viera obligado a adquirir un producto nuevo igual o similar y cubrir esa necesidad. Esto influye mucho en el desarrollo de la economía, ya que asegura una gran demanda, donde las empresas tienen más beneficios y una continua oferta. ([Ortega, O. Programa de Renovación y Actualización de máquinas \(RAM\), UNC 2021](#)).

²⁵ La sustentabilidad en el marco de la Ley de educación ambiental, se define como un proyecto social que integra el desarrollo con justicia social, la distribución de la riqueza, la preservación de la naturaleza, la igualdad de género, la protección de la salud, la democracia participativa y el respeto por la diversidad cultural, y que busca el equilibrio entre diversas dimensiones, como la social, la ecológica, la política y la económica, en el marco de una ética que promueve una nueva forma de habitar nuestra casa común. Ley 27.621. [InfoLEG - Ministerio de Justicia y Derechos Humanos - Argentina](#).

B. Área Programación

El aprendizaje de la programación consiste en conocer cómo se desarrollan las soluciones de software para diversos problemas. El proceso de desarrollo para crear programas claros, significativos y eficientes implica elegir qué procesar, cómo procesarlo y cómo almacenar información, dividir los problemas grandes en otros más pequeños, recombinar las soluciones existentes y analizar diferentes soluciones.

Se incluye un eje sobre representación de la información en el área de programación para tener una mirada más compleja del área usualmente abordada desde las acciones. En los enfoques más comunes, los datos se utilizan pero pocas veces se conceptualizan. Comprender cómo “aparecen” los datos al momento de programar tiene importancia para comprender el rol humano en la manipulación de estos y la incidencia de sus características en la infraestructura tecnológica.

La propuesta aborda la programación con un doble propósito: por un lado, en tanto práctica fundamental para que las y los estudiantes sean capaces de construir sus propias soluciones computacionales y, por otro, dado que el software es un componente imprescindible de todos los dispositivos y artefactos computacionales, conocer cómo se construye es fundamental para completar los conceptos de computación trabajados en las demás áreas.

Eje B1. Soluciones a problemas computacionales

La solución a problemas computacionales implica el análisis de los problemas, la definición de una estrategia y su expresión mediante la utilización de diversas técnicas y herramientas que facilitan la tarea, y que son independientes del lenguaje de programación que se utilice. Conocer y dominar técnicas para la solución de problemas computacionales, e identificarlas y analizarlas durante la construcción de programas apunta a que las y los estudiantes comprendan mejor este proceso, lo dominen desde su concepción hasta su finalización en la forma de un programa y sean capaces de construir programas legibles, reutilizables y modulares²⁶. Dado que los algoritmos²⁷ son formas sofisticadas de resolver problemas, se abordan en el último nivel vinculados a nociones de teoría de la computación como la complejidad y la computabilidad.

²⁶ Un programa es modular si está construido basándose en módulos o subprogramas. Un módulo es cada una de las partes de un programa que resuelve uno de los subproblemas en que se divide el problema complejo original. Cada uno de estos módulos tiene una tarea bien definida y algunos necesitan de otros para poder operar.

²⁷ En esta propuesta consideramos que un algoritmo es una descripción general de un proceso que permite construir una solución para todas las instancias de un problema, por ejemplo, cómo transformar una lista cualquiera de números para que quede ordenada de menor a mayor. Esta concepción es mucho más sofisticada que una secuencia de pasos, por eso en esta propuesta no consideramos como algoritmos los pasos para resolver tareas cotidianas (como atarse los cordones).

En la siguiente tabla se describen los alcances previstos para cada uno de los ciclos escolares en relación a los temas que pertenecen a este eje.

Área →	Programación		
Eje →	Soluciones a problemas computacionales		
Saberes →	Primaria - 2º Ciclo	Secundaria - Ciclo Básico	Secundaria - Ciclo Orientado
	<p>La resolución de problemas computacionales a través del diseño de estrategias que permitan estructurar y construir programas, identificando elementos repetitivos y regularidades en el problema, expresando acciones mediante división en subtareas y expresando estas como procedimientos con denominaciones adecuadas, siempre en vinculación con las herramientas específicas de lenguajes de programación. La comprensión de que el programa producido es una forma de comunicar la solución propuesta a personas y máquinas.</p>	<p>La resolución de problemas computacionales a través del diseño de estrategias que permitan estructurar y construir programas, incorporando la dimensión de los datos (representación de datos y transformaciones sobre ellos), siempre en vinculación con las herramientas específicas de lenguajes de programación. La comprensión de características importantes en el proceso de programación, como las capas de abstracción, la legibilidad, y otras que contribuyen a expresar mejor las soluciones.</p>	<p>El desarrollo de criterios para evaluar e implementar soluciones desde distintos puntos de vista (legibilidad, modularidad, reutilización y eficiencia) teniendo en cuenta las necesidades del problema. La comprensión de nociones de computabilidad y complejidad, el reconocimiento de los algoritmos y estructuras de datos como soluciones computacionales generales.</p>
Temas ↓	Alcances ↓		
	Primaria - 2º Ciclo	Secundaria - Ciclo Básico	Secundaria - Ciclo Orientado

1.

Diseño de soluciones computacional es: estrategias de solución, modularidad y legibilidad.

Identificar, diferenciar y comprender situaciones problemáticas donde haya intervenido una **solución computacional** o parte de ella y en particular sistemas que incluyan componentes de software para cómputo.

Identificar características relevantes y **comportamientos repetidos** de una situación problemática claramente enunciada y con límites precisos, para descomponer en términos de **subproblemas** asignándoles una **denominación representativa** (técnica de división en subproblemas).

Utilizar la técnica de división de un problema en subproblemas tanto para pensar y diseñar soluciones a problemas como para programar mediante el uso de **procedimientos**²⁸.

Reflexionar sobre el hecho de que un **programa** es una manera de **comunicar**

Adquirir fluidez en el uso de la técnica de división de subproblemas para elaborar **estrategias** de solución, y crear programas utilizando las herramientas del lenguaje disponibles.

Considerar los **datos** involucrados en un problema como parte de la estrategia de solución.

Reconocer las **interfaces de programación** como el mecanismo para acceder a un conjunto de procedimientos y funciones que definen lo que se puede hacer y conocer sobre una entidad o aspecto del problema sin necesariamente conocer las soluciones (por ejemplo, los bloques que nos provee la extensión del lápiz en Scratch es su interfaz, y las **bibliotecas de funciones** poseen una interfaz que permite trabajar con ellas).

Comprender las interfaces de programación como una herramienta que

Adquirir fluidez en el uso de la técnica de división de subproblemas para elaborar estrategias de solución, y crear programas y definir bibliotecas³¹ utilizando las herramientas del lenguaje disponibles.

Incorporar la noción de **modularización del código** al diseño de soluciones, a través de la definición de procedimientos y funciones y de bibliotecas de código.

Extender la noción de **legibilidad** al uso y definición de bibliotecas y tipos de datos estructurados.

Utilizar la modularización y legibilidad para diseñar soluciones **reutilizables**.

Comparar diferentes bibliotecas para decidir cuál es conveniente para el problema planteado.

²⁸ Ver el eje B3. Lenguajes de programación.

³¹ En computación, una biblioteca (del inglés *library*, popularizado también como "librería") es un módulo de software que ofrece una interfaz bien definida para interactuar con un conjunto de operaciones vinculadas temáticamente. A diferencia de un programa, no se espera que la biblioteca sea utilizada en forma autónoma, sino que su fin es ser utilizada en otros programas para resolver problemas específicos sin conocer los detalles de la implementación de las soluciones. A través del uso de bibliotecas bien definidas es posible construir programas modularmente de forma mucho más eficaz.

soluciones a problemas computacionales entre personas que programan, y que, al mismo tiempo, debe servir para hacer funcionar a una máquina que solucione el problema al ejecutar el código.

Diferenciar características de un problema que se mantienen **fijas de aquellas que varían** en las diferentes instancias del problema y organizar la estrategia de solución sobre la base de ello (por ejemplo, identificar que en un laberinto la cantidad de casillas es constante, pero la posición de los obstáculos es variable entre las distintas ejecuciones).

Reconocer la necesidad de conservar, acceder y modificar información de contexto (por ejemplo, el puntaje actual en un juego).

Reconocer la existencia de **soluciones previas** que pueden ser utilizadas como subtareas en una solución propia sin conocer cómo están implementadas.

Identificar **regularidades en un programa** y expresarlas a través de las herramientas

permite **reutilizar** procedimientos y funciones predefinidos y utilizarlos sin conocer cómo fueron desarrollados, pero comprendiendo su utilidad, por lo que constituyen una primera aproximación de la noción de **capa de abstracción**.

Entender la noción de **legibilidad de un programa** como la cualidad de un programa que permite que otra persona, al leer el código, pueda comprender la solución propuesta (por ejemplo, que el nombre elegido para una variable describa la intención del dato que debe guardarse allí).

Diseñar **programas legibles** definiendo procedimientos, funciones, variables y parámetros para expresar las subtareas y los datos encontrados en el problema, asignándoles una denominación representativa.

Incorporar, al momento de pensar una solución, la posibilidad de **modificar un programa existente** para adecuarlo a problemas similares, y comprender que la legibilidad facilita las posibilidades de modificación.

Discernir en qué **nivel de abstracción** trabajar en función del problema a resolver (biblioteca para trabajar sobre videos, o utilizar una biblioteca que trabaje directamente sobre píxeles).

Estimar la **eficiencia** de un programa en términos de la cantidad de operaciones y de consumo de memoria.

Reconocer si una solución computacional es implementable teniendo en cuenta las limitaciones de eficiencia del dispositivo computacional.

Desarrollar criterios para evaluar soluciones desde distintos puntos de vista (legibilidad, reutilización y eficiencia) teniendo en cuenta las necesidades del problema.

Extender la noción de programa correcto a sus procedimientos, funciones o módulos.

adecuadas del lenguaje (por ejemplo, repeticiones y procedimientos).

Elaborar, a partir de un problema dado claramente enunciado y con límites precisos, una **estrategia** en términos del dominio del problema²⁹ (sus características relevantes, fijas y variables, subproblemas, comportamientos repetidos, soluciones previas), y conocer las herramientas provistas por el lenguaje de programación.

Construir la noción de un **programa correcto**, teniendo en cuenta si los resultados obtenidos al ejecutarlo son los esperados con respecto al problema.

Identificar **regularidades** en un programa y expresarlas a través de las herramientas adecuadas del lenguaje (por ejemplo, repeticiones, procedimientos y funciones y **parámetros**³⁰) para mejorar su legibilidad.

²⁹ El dominio de un problema son los elementos que integran el problema y las relaciones que existen entre ellos en el nivel del problema. Por ejemplo, si se está programando una aplicación bancaria, el dominio habla de “cuentas bancarias”, de “depositar en una cuenta”, de “extraer de una cuenta”, de “saldo de una cuenta”, etc. Luego, según cómo se elige representar las cuentas, las operaciones se implementan en términos de la representación. Entonces, si para una cuenta se elige guardar dos datos (tipo y número), la operación de depositar va a incrementar el número, la de extraer verificará si el número es mayor que cero en una caja de ahorros, o mayor que cierto “descubierto” en una cuenta corriente, y restará, etc.

³⁰ Ver el eje B3. Lenguajes de programación.

2.

Algoritmos y estructura de datos:
problemas clásicos, complejidad y recursión.

Comprender que un algoritmo es una solución general que funciona para todas las instancias de un problema en forma automática, reconociendo la existencia de diferentes formas algorítmicas (dividir y conquistar, algoritmos golosos, programación dinámica, algoritmos heurísticos, etc.), para ampliar la noción de algoritmo como forma técnica en el abordaje de la solución de problemas.

Reconocer la necesidad de la generalidad al momento de pensar una solución de forma algorítmica (cómo elaborar una solución que funcione para todas las instancias de un mismo problema).

Reconocer que el **diseño de un algoritmo está limitado** a las posibilidades de ejecución en una computadora (tanto respecto de sus formas de operación como respecto de la cantidad de datos a procesar).

Comprender las **estructuras de datos** como formas de organizar los datos desde el punto de vista de las operaciones que se requieren para acceder a ellos, considerando la eficiencia de los accesos en

			<p>forma general (por ejemplo, no es lo mismo tener datos desordenados que ordenados, no es lo mismo acceso lineal que binario, etc.).</p> <hr/> <p>Conocer la noción de complejidad temporal y espacial³² de un algoritmo, y utilizarla para comparar la eficiencia de diferentes algoritmos.</p> <hr/> <p>Comprender la influencia de la manera de organización de los datos (o sea, de las estructuras de datos) en la definición y funcionamiento de algoritmos efectivos y en el costo final de los mismos (por ejemplo, árboles binarios para realizar búsquedas binarias).</p> <hr/> <p>Conocer la noción de recursión como una forma de resolver problemas computacionales basándose en instancias más pequeñas del mismo problema (por ejemplo, ordenar una secuencia de elementos dividiéndola en dos partes iguales, ordenarlas de la misma manera y</p>
--	--	--	--

³² La complejidad de un algoritmo es una medida estimativa del consumo esperado de recursos del algoritmo al ejecutarlo en una entrada desconocida. Se habla de complejidad temporal cuando se hace referencia a la cantidad de trabajo requerida (usualmente vinculada al tiempo que demora el cálculo) y de complejidad espacial cuando se hace referencia a la cantidad de información que se debe manipular (usualmente vinculada al espacio de almacenamiento necesario).

Existen numerosas formas de calcular la complejidad. La más sencilla es contar la cantidad de milisegundos o bytes requeridos por una instancia particular, aunque es una medida poco fiel para casos genéricos, por lo que existen formas más sofisticadas que generalizan muchos casos particulares.

			<p>luego combinar las dos soluciones en el resultado final).</p>
<p>3.</p> <p>Teoría de la computación: factibilidad y computabilidad</p>			<p>Entender la noción de computabilidad de un problema como la existencia de una solución algorítmica para resolverlo.</p> <hr/> <p>Identificar que existe un límite para los problemas que se pueden resolver con una computadora y conocer problemas no computables.</p> <hr/> <p>Entender la noción de factibilidad de un algoritmo como un límite sobre su complejidad.</p> <hr/> <p>Conocer que existen problemas inherentemente difíciles (es decir, problemas computables, pero para los cuales no se conoce una solución factible).</p> <hr/>

			<p>Conocer la existencia de una clasificación de problemas según su complejidad (clases P, NP, NP-Completo y las relaciones entre ellas).</p>
--	--	--	---

Eje B2. Representación de información en la resolución de problemas computacionales.

La computación puede entenderse como el estudio del procesamiento de la información. Para poder ser procesada, la información debe ser primero representada dentro del sistema de cómputo, que luego manipulará dichas representaciones para lograr el procesamiento. Además, las características de la representación, tales como la eficiencia en cantidad de símbolos o la legibilidad, tienen una influencia central en las posibilidades del cómputo, y habilitan aplicaciones tales como la compresión de datos o la criptografía.

Por esta razón, incorporar este eje a las prácticas de enseñanza permite que las y los estudiantes se aproximen al concepto de representación de información y sus características y sean capaces de expresar información mediante representaciones en sus programas y trabajo con datos. Además, la reflexión desde este enfoque, permitirá identificar estas nociones fundamentales presentes en todos los procesos de cómputo.

En la siguiente tabla se describen los alcances previstos para cada uno de los ciclos escolares en relación a los temas que pertenecen a este eje.

Área →	Programación		
Eje →	Representación de información en la resolución de problemas computacionales		
Saberes →	Primaria - 2º Ciclo	Secundaria - Ciclo Básico	Secundaria - Ciclo Orientado
	El reconocimiento de la representación de información en diferentes situaciones y contextos para poder utilizar herramientas de lenguajes de programación que representen información (como sensores ³³).	El reconocimiento de la existencia de capas de representación que establecen diferentes niveles de abstracción para utilizar diferentes herramientas del lenguaje en la definición de nuevas representaciones (mediante tipos básicos ³⁴ y registros ³⁵) y modelizar la información de diferentes dominios cotidianos ³⁶ . La comprensión de la computación como el estudio del procesamiento de la información, entendiéndolo que toda la información debe ser representada para poder procesarla.	La comprensión del papel que juegan la eficiencia y la legibilidad en el diseño de las representaciones que se utilizan incluyendo la compresión y la criptografía. La modelización de información de diferentes dominios cotidianos mediante herramientas de los lenguajes de programación (como tipos básicos, registros, listas y sus combinaciones).
Temas ↓	Alcances ↓		
	Primaria - 2º Ciclo	Secundaria - Ciclo Básico	Secundaria - Ciclo Orientado

³³ Ver el eje B3 de Lenguajes de programación.

³⁴ Los tipos básicos de un lenguaje de programación son aquellos que ya vienen contruidos con el lenguaje. Usualmente incluyen a los números y los booleanos, y pueden incluir otros como cadenas de caracteres (Strings), colores, direcciones, etc.

³⁵ Ver el eje B3 de Lenguajes de programación.

³⁶ Un dominio cotidiano es un dominio con el que se encuentran todos los días, como puede ser una red social, la agenda del celular, la lista de asistencia del aula, etc.

1.

Nociones de representación de información.

Identificar **situaciones** en las que se **representan entidades mediante datos** (por ejemplo, utilizar un número 1 para representar una manzana y un número 2 para representar una banana, u otras representaciones sencillas, tales como representar puntos cardinales con números, estados del mar con banderas de colores, colores en los semáforos, etc.).

Reconocer que la **representación de una entidad es arbitraria** y la definen las personas en un contexto (por ejemplo, los antiguos romanos representaban los números de una forma y nosotros de otra).

Reconocer las herramientas de lenguajes de programación que permiten **expresar información** (sensores y expresiones literales³⁷) y utilizarlas en operaciones sencillas (detección de condiciones simples, operaciones aritméticas simples).

Comprender que la información de un sistema computacional requiere ser representada mediante elementos de ese sistema (por ejemplo, números, cadenas de caracteres, cadenas de bits, etc.) para poder procesarse, y que el procesamiento consiste en la manipulación de dichas representaciones.

Comprender que la computación es el estudio del procesamiento de la información y que al utilizar cualquier artefacto computacional estarán manipulando datos.

Reconocer la existencia de representaciones de entidades mediante otras representaciones, que constituyen una serie de **capas de representación** (por ejemplo, un video se representa como una secuencia de imágenes, y esas imágenes se representan como secuencias de colores, y los colores como grupos de números y finalmente los números se pueden representar de forma binaria a través del uso de corriente en una computadora).

Reconocer la existencia de representaciones de entidades mediante **expresiones y tipos de datos**, incluyendo representaciones que involucren secuencias de datos con estructura. Por ejemplo, la lista de estudiantes modelados como un registro con nombre y cantidad de días que faltaron.

Reconocer las **herramientas de lenguajes** de programación que permiten expresar información (sensores, expresiones literales y funciones sobre tipos de datos tales como listas y registros para modelar entidades y operaciones del mundo cotidiano) y utilizarlas para construir transformaciones de información (sensores que combinan otros sensores y condiciones básicas sobre representaciones elementales, como comparación de valores).

Comprender que la misma información se puede representar de diferentes maneras y ello impacta en la **eficiencia** (cantidad de elementos y de tiempo de cómputo, viabilidad de las transformaciones) y en la **legibilidad** (caracterización de la entidad a

³⁷ Ver el eje B3. Lenguajes de programación.

		<hr/> <p>Reconocer las herramientas de lenguajes de programación que permiten expresar información (sensores, expresiones literales y funciones) y utilizarlas para construir transformaciones de información (sensores que combinan otros sensores y condiciones básicas sobre representaciones elementales, como comparación de valores).</p> <hr/> <p>Comprender que las transformaciones posibles de la información están vinculadas al tipo de información y relacionarlo con la noción de tipo de datos en los lenguajes de programación.</p> <hr/> <p>Reconocer diferencias entre la representación digital y la analógica³⁸.</p>	<p>representar). Por ejemplo, una representación de números usando porotos en forma unaria requiere 1 millón de porotos para representar 1 millón de puntos, mientras que una representación posicional requiere solamente 7 símbolos.</p> <hr/> <p>Vincular las formas de representación de bajo nivel (numeración binaria) con la idea de representación digital y con las características de eficiencia, legibilidad y las capas de representación vistas, y comprender las características que la hacen una buena representación (eficiencia en la cantidad de símbolos utilizados haciendo factible su implementación física en gran escala).</p> <hr/> <p>Reconocer la compresión como una obtención de representaciones más económicas en cantidad de elementos usados para representar, con fines de mejorar el funcionamiento de un sistema computacional.</p> <hr/>
--	--	--	---

³⁸ Las representaciones analógicas utilizan una forma continua de señal como forma de codificación; estas representaciones usualmente están relacionadas con alguna forma física de representar la información. Las representaciones digitales, en cambio, emplean valores discretos (por ejemplo, una cantidad fija de números) para codificar la información. Por ejemplo, una representación analógica del sonido representa a este como una variación continua y constante de fluctuaciones de voltaje de acuerdo a las variaciones de presión de aire hechas por el sonido. En cambio, una representación digital del mismo sonido emplea una cantidad discreta de valores fijos de voltaje medidos cada cierto intervalo muy corto de tiempo.

			<p>Reconocer la criptografía como una obtención de representaciones de difícil interpretación sin tener las claves necesarias incluso por sistemas computacionales.</p>
<p>2. Representaciones estandarizadas (imágenes, sonido, textos y números).</p>	<p>Conocer la forma en que se representan vídeos, imágenes y textos (por ejemplo, un video representado como una secuencia de imágenes, una imagen como una cuadrícula de píxeles y que cada píxel se representa con tres números, un texto representado como secuencia de caracteres, y cada carácter representado como un número).</p>	<p>Comprender la noción de estándar de representación y conocer los más frecuentes para las entidades abordadas (imágenes como mapa de bits tales como BMP o GIF, caracteres a través de formas de codificación de caracteres tales como ASCII o UNICODE, sonidos como secuencias de muestras —PCM— tales como WAV o AIFF).</p>	<p>Reconocer que los formatos comprimidos (por ejemplo, H265, JPG/PNG, MP3/FLACC) están diseñados para un uso más eficiente de los recursos.</p> <hr/> <p>Reconocer que los videos necesitan de formatos comprimidos para poder ser procesados, compartidos, almacenados.</p> <hr/> <p>Valorar los estándares abiertos por sobre los cerrados en términos de distribución y acceso.</p>

3.

Modelado:
representación
con datos
estructurados

Asociar la **práctica de modelado** con la identificación de atributos que sean significativos para el problema, de las entidades que se quieren expresar en el programa. Por ejemplo, si la entidad considerada es una persona humana, sus atributos podrían ser el género, la edad, la profesión, el lugar de residencia, el nombre, el número de DNI, etc.

Reconocer los **datos con estructura**, en particular registros definidos por otras personas, como modelos que representan entidades del mundo cotidiano, tales como cartas o fechas.

Interpretar modelos conformados por registros de modo que permitan inferir la entidad que se está modelando.

Interactuar con **registros** para operar con las entidades que representan (por ejemplo, si modelamos un botón como un registro que contiene como atributos texto, color y tamaño de letra, para modificar el color hay que cambiar el valor del atributo color).

Identificar situaciones en las que hay entidades que se pueden **representar con listas** (por ejemplo, los amigos de una cuenta en una red social se representan con una lista de registros de usuarios, un diccionario se representa con una lista de asociaciones entre palabras y definiciones).

Comprender que cuando un programa trabaja con secuencias trata los elementos de a uno por vez, a diferencia de las personas, que obtenemos información de una secuencia considerando todos los elementos al mismo tiempo (por ejemplo, al mirar una mano de cartas podemos determinar si son todas del mismo palo o no).

Definir **registros** para modelar entidades con atributos.

Utilizar **listas** para modelar secuencias o conjuntos de entidades.

			<div data-bbox="1563 156 2101 347" style="background-color: #d4edda; padding: 10px;"> <p>Combinar registros y listas para modelar secuencias y conjuntos de entidades con atributos.</p> </div>
--	--	--	---

Eje B3. Lenguajes de programación

La programación es uno de los pilares de la computación, en tanto permite expresar y materializar el procesamiento de información al ejecutar los programas en una máquina. Los lenguajes de programación proveen las herramientas que permiten escribir los programas.

Solo se puede programar utilizando un lenguaje de programación. El propósito de este eje es proveer un marco general para la enseñanza de los lenguajes de programación a partir de nociones generales comunes que permitan comprender algunas herramientas centrales y transferibles entre lenguajes asociadas a los problemas que permiten resolver. Este enfoque permite correr el foco de las particularidades de un lenguaje.

La propuesta define un conjunto de herramientas de lenguajes de programación elegidas por su centralidad y generalidad. Se propone iniciar utilizando lenguajes de bloques, pues permiten focalizar en los conceptos y no lidiar con la sintaxis de los lenguajes textuales. A medida que se avanza en la escolaridad, se incorporan nuevos conceptos y herramientas y se propicia la transferencia de los conocidos a diferentes lenguajes. Este recorrido termina en los lenguajes textuales para aproximarse a las herramientas empleadas en la construcción de los programas con los que interactúan las personas en su vida contemporánea y para poder aprovechar una mayor cantidad de recursos disponibles para construir programas propios (en particular para poder utilizar los recursos de programación colaborativos como las bibliotecas).

En la siguiente tabla se describen los alcances previstos para cada uno de los ciclos escolares en relación a los temas que pertenecen a este eje.

Área →	Programación		
Eje →	Lenguajes de programación		
Saberes →	Primaria - 2º Ciclo	Secundaria - Ciclo Básico	Secundaria - Ciclo Orientado
	La comprensión de conceptos generales de lenguajes de programación y la construcción de programas utilizando las herramientas propuestas para este ciclo (alternativas, repeticiones, procedimientos, expresiones con sus tipos y eventos ³⁹).	La comprensión de conceptos generales de lenguajes de programación y la construcción de programas sumando las herramientas propuestas para este ciclo (parámetros, funciones, uso de registros) y la transferencia de conceptos entre diferentes lenguajes.	La comprensión de conceptos generales de lenguajes de programación y la construcción de programas sumando las herramientas propuestas para este ciclo (definición de listas y registros, bibliotecas y servicios ⁴⁰) y la transferencia de conceptos entre diferentes lenguajes.
Temas ↓	Alcances ↓		
	Primaria - 2º Ciclo	Secundaria - Ciclo Básico	Secundaria - Ciclo Orientado
1. La sintaxis como un aspecto de los lenguajes de programación en bloques y texto.	Reconocer que existen reglas para la combinación de bloques que determinan cuáles son las combinaciones válidas.	Reconocer que existen reglas para la combinación de comandos y expresiones que determinan cuáles son las combinaciones válidas tanto en lenguajes bloques como en lenguajes textuales.	Conceptualizar la sintaxis como las reglas de combinación de comandos y expresiones que determinan las combinaciones válidas.

³⁹ Ver el eje B3 de Lenguajes de programación por definiciones de todas estas herramientas.

⁴⁰ Un servicio es una aplicación que realiza tareas automatizadas y brinda cierta información en forma remota a otros programas (por ejemplo, conocer la hora oficial o el valor presente del tipo de cambio entre dos monedas).

			<p>Comprender que la sintaxis en lenguajes de bloques está dada por la forma de los bloques, por lo que es muy difícil encontrar errores de sintaxis.</p> <hr/> <p>Comprender que en los lenguajes textuales aparece una nueva categoría de errores asociada a la sintaxis.</p> <hr/> <p>Utilizar adecuadamente los símbolos según la sintaxis del lenguaje para construir programas utilizando texto.</p>
<p>2.</p> <p>La semántica como el significado de los programas y sus partes en términos del problema que resuelve.</p>	<p>Anticipar qué modificaciones del escenario produce (qué hace) un programa en un entorno de enseñanza para la programación sin necesidad de ejecutarlo.</p> <hr/> <p>Construir el significado de un procedimiento como modificación</p>	<p>Anticipar qué transformación de información expresa el uso de una función con argumentos⁴¹ en un entorno de enseñanza de programación sin necesidad de ejecutarla.</p> <hr/> <p>Construir el significado de una función con parámetros como una transformación de información (la información de los argumentos</p>	<p>Conceptualizar la semántica de un programa como el significado en términos del problema que resuelve.</p> <hr/> <p>Identificar que dos programas que resuelven el mismo problema son equivalentes semánticamente, aún</p>

⁴¹ El uso de función con argumentos es un caso de utilización de una función (ver el eje B3. Lenguajes de programación) que toma datos específicos para sus parámetros, y que describe el resultado de transformar esos datos en el resultado, según la definición dada.

que se produce en el escenario al ejecutarlo, distinguiéndolo de la secuencia de pasos involucrados en tal modificación. Por ejemplo, el contraste entre “ir a la puerta” y la secuencia “mover 4 pasos a la derecha, y luego mover 3 pasos adelante”.

Comprender que la definición de un procedimiento incorpora un nuevo comando al lenguaje.

Reconocer la diferencia entre **comandos y expresiones**, en tanto descripciones de acciones y datos, respectivamente.

— datos de entrada— se transforma en el resultado de la función —datos de salida—, distinguiéndolo de las operaciones involucradas en dicha transformación. Por ejemplo, el contraste entre transformar la información de notas obtenidas en diferentes materias en el promedio de dichas notas y sumar las notas de a una y dividir por la cantidad de notas sumadas.

Comprender que la definición de una función incorpora un nuevo operador al lenguaje, y que el uso de ese operador aplicado a argumentos específicos es diferente de la definición de la función.

Interpretar la **aplicación de una función** a argumentos específicos como un valor (resultado de la operación) y no como el proceso involucrado en su cálculo. Por ejemplo, interpretar “el promedio de mis notas” como un número y no como un proceso de cálculo.

Identificar los **procedimientos** como representación de subtareas en el mundo de los comandos y las funciones como representación de subproblemas en el mundo de las expresiones.

cuando su sintaxis sea diferente o incluso estén escritos en diferentes lenguajes.

Apreciar la importancia de comprender los programas desde un punto de vista semántico para saber lo que resuelven y poder reutilizarlos en diferentes contextos y corregir sus errores, completando así la visión de la programación como la comunicación de soluciones a problemas computacionales que además sean efectivos para su utilización práctica.

Reconocer el rol de la **documentación** en la tarea de programar para facilitar y completar el aspecto comunicacional de los programas.

		<p>Identificar que tanto los procedimientos como las funciones extienden el lenguaje con nuevas herramientas específicas (comandos y operadores).</p> <p>Comprender que la solución de un problema computacional expresada por un programa se construye a partir de la combinación de transformaciones de estado (modificaciones del escenario) y transformaciones de información⁴².</p>	
<p>3. Herramientas de lenguaje de programación.</p>	<p>Crear programas en entornos de enseñanza de programación por bloques que combinen comandos, expresiones⁴³ y eventos⁴⁴ para resolver problemas.</p>	<p>Crear programas en entornos de enseñanza de programación por bloques o texto que combinen comandos, expresiones y eventos para resolver problemas.</p> <ul style="list-style-type: none"> ● Utilizar comandos primitivos, secuencias, alternativas 	<p>Crear programas en entornos de enseñanza de programación en texto o en lenguajes textuales que combinen comandos, expresiones y eventos para resolver problemas.</p>

⁴² Existen dos maneras de expresar soluciones computacionales: transformando información (por ejemplo, calcular un promedio de notas) o transformando un estado (por ejemplo, usar un brazo robótico para mover un grupo de cajas de un estante a un vehículo o viceversa).

⁴³ Una expresión es una herramienta de un lenguaje de programación que permite describir información.

⁴⁴ Un evento en un lenguaje de programación es un mecanismo para disparar la ejecución de cierta parte del código basándose en la ocurrencia de cierta situación en el entorno de trabajo. Ejemplos de situaciones que disparan un evento son el clic de una *mouse*, el apretar una tecla, girar el celular, el paso del tiempo (sobre la base de los tictacs de un reloj), etc.

- Utilizar **comandos** primitivos, secuencias, alternativas condicionales⁴⁵, repeticiones simples⁴⁶ y procedimientos en la construcción de programas.
- Utilizar **expresiones** literales⁴⁷ (números y cadenas de caracteres⁴⁸ – *strings*), sensores y operadores simples⁴⁹ en la construcción de programas.
- Utilizar **eventos** de interacción con el usuario (por ejemplo, para responder al uso de teclas o del *mouse*) para definir programas interactivos.

condicionales, repeticiones simples, repeticiones condicionales y procedimientos en la construcción de programas.

- Utilizar **expresiones** literales (números, cadenas de caracteres y booleanos⁵¹), sensores, parámetros, variables⁵², registros, funciones y operadores simples en la construcción de programas.
- Utilizar **eventos** de interacción con el usuario o entorno (por ejemplo, para responder al uso de teclas o del mouse o responder a un hardware específico como el reloj o el acelerómetro) para definir programas interactivos.

- Utilizar **comandos** primitivos, secuencias, alternativas condicionales, repeticiones simples, repeticiones condicionales, repeticiones indexadas⁵⁷ y procedimientos en la construcción de programas.
- Utilizar **expresiones** literales (números, cadenas y booleanos), sensores, parámetros, variables, registros, listas, funciones y operadores simples en la construcción de programas.
- Utilizar **eventos** de interacción con el usuario o entorno (por ejemplo, para responder al uso de teclas o del mouse o responder a un hardware específico como el reloj o el acelerómetro) para definir programas interactivos.

⁴⁵ Una alternativa condicional es una forma de combinar comandos o expresiones en un lenguaje de programación que permite elegir entre dos comandos o expresiones, según corresponda, basándose en una condición. Por ejemplo, basándose en un sensor que indique si hayUnObstáculoDelante, elegir entre IrHaciaAdelante o RodearElObstáculo.

⁴⁶ Una repetición es una forma de combinar comandos en un lenguaje de programación que permite ejecutar la misma secuencia de comandos más de una vez. Existen formas de repetición simple, basadas en una cantidad fija de repeticiones, o condicional, basadas en una condición que debe cumplirse para poder terminar de repetir.

⁴⁷ Una expresión literal es una expresión que menciona literalmente al valor que describe. También se suele hablar de “un literal” para hablar de una expresión literal. Por ejemplo, los literales numéricos son 0, 1, 2, 3, etc. En contraposición, el sensor kmALaCiudad también es una expresión numérica, pero no literal, porque no se sabe su valor hasta no estar ejecutando.

⁴⁸ El tipo cadena de caracteres (en inglés, string) es un tipo de datos que representa secuencias de letras, y que se pueden utilizar para representar palabras, oraciones o secuencias arbitrarias de caracteres sin restricciones.

⁴⁹ Un operador es una expresión que describe la transformación de uno o más datos dados como argumentos en una información resultado. Los operadores simples son operadores básicos, como la suma, la multiplicación, la negación, etc.

⁵¹ El tipo booleano es un tipo de datos que representa valores de verdad. Los valores booleanos son dos: verdadero y falso. El nombre de este tipo de datos se basa en el nombre del matemático que estudió la matemática asociada a los valores de verdad y sus operaciones: George Boole. Este tipo de datos se utiliza en alternativas y repeticiones condicionales como forma de tomar decisiones (cuál alternativa elegir, o si continuar repitiendo o frenar).

⁵² Una variable de un lenguaje de programación es un mecanismo para recordar información a través de un nombre. En algunos lenguajes estos nombres se asocian con posiciones de memoria, mientras que en otros tienen más similitudes con las variables de un problema matemático.

⁵⁷ Una repetición indexada es una forma de repetición que utiliza un índice para repetir la secuencia de comandos indicada para cada elemento de una lista dada (en forma explícita o implícita).

Identificar las herramientas de lenguajes de programación aprendidas en los diferentes entornos con los que trabajan, reconociendo que estas son **independientes del lenguaje** específico utilizado o del “estado del arte” de la tecnología.

Definir y utilizar **procedimientos sin parámetros**⁵⁰ para plasmar estrategias de solución en el programa expresando subtareas.

Utilizar expresiones y comandos para **conservar, acceder y modificar información** de contexto (por ejemplo, el puntaje actual de un juego) durante la ejecución de un programa implementados de alguna manera (por ejemplo, con datos en el entorno y expresiones primitivas para acceder y comandos primitivos para modificar).

Comprender que las herramientas de lenguajes de programación aprendidas son independientes del lenguaje específico utilizado o del “estado del arte” de la tecnología, y ser capaces de **reconocerlas en diferentes lenguajes**, para poder explorar nuevos lenguajes por cuenta propia.

Comprender que muchas operaciones primitivas (tanto comandos como expresiones) requieren que se les suministre información para su funcionamiento, y que la forma de hacerlo es a través del uso de expresiones en forma de **argumentos** en su invocación.

Definir y utilizar **funciones**⁵³ para expresar de manera sencilla y legible expresiones más complejas.

Identificar la necesidad y utilidad de los **parámetros**⁵⁴ como formas de proveer información en la forma de argumentos a los procedimientos y funciones que se definen.

Reconocer que las **bibliotecas** son una manera de agregar funcionalidades a las herramientas básicas de un lenguaje, incorporarlas a sus programas e invocar las operaciones que proveen.

Utilizar algún lenguaje de **programación textual**⁵⁸, para enfrentarse con problemas no resueltos en entornos que no son de enseñanza, transponiendo las herramientas ya aprendidas incluso cuando aparecen combinadas con otras o modificadas de alguna forma.

Comprender que los conceptos de programación aprendidos no están ligados a un lenguaje en particular y que, por lo tanto, son independientes del lenguaje específico utilizado o del “estado del arte” de la tecnología.

⁵⁰ Un procedimiento es una herramienta de lenguajes de programación que permite definir un nuevo comando en el lenguaje.

⁵³ Una función es una herramienta de lenguajes de programación que permite definir una nueva expresión en el lenguaje.

⁵⁴ Un parámetro en un procedimiento o función es un agujero al que se le da nombre, y que espera se provea un cierto dato (el argumento) que se debe proveer al usar la operación, para poder completar la definición y funcionar como se espera. Por ejemplo, en la operación de EscribirNombre, los parámetros podrían ser nombreAEscribir y colorAUtilizar.

⁵⁸ En este ciclo se propone utilizar un lenguaje textual como una forma de acercar a las y los estudiantes a experiencias de programación fuera de entornos didácticos.

Reconocer que los datos involucrados en un programa pueden ser de distinto tipo (por ejemplo, números, letras, colores, cadenas) y que el **tipo de dato** está relacionado con cómo se puede operar con ellos (por ejemplo, los números se pueden sumar, las palabras se pueden concatenar).

Considerar los tipos de datos involucrados en los argumentos de un comando para interpretar y solucionar errores de ejecución.

Reconocer la importancia de construir programas que sean entendibles por las personas y utilizar las herramientas brindadas por el lenguaje de programación para conseguirlo (en particular, uso de procedimientos para la división en subtareas y denominación representativa de los identificadores).

Incorporar **parámetros** a la definición de procedimientos y funciones de manera de expresar soluciones generales a problemas con mayor número de instancias mediante un único procedimiento o función.

Reflexionar sobre la utilización de procedimientos y funciones con parámetros como forma de proveer mecanismos de **reutilización y modularización**.

Definir y utilizar **procedimientos y funciones con y sin parámetros** para plasmar estrategias de solución en el programa explicitando los subproblemas.

Reconocer que las **expresiones** de un programa tienen un **tipo de dato**⁵⁵ (por ejemplo, números, letras, colores, cadenas, booleanos, registros) que está relacionado con cómo se puede operar con ellas (por ejemplo, las expresiones de tipo “número” se pueden sumar, las expresiones de tipo “cadena” se pueden concatenar, las expresiones de tipo “booleano” se pueden combinar mediante negaciones, conjunciones y disyunciones, y los

Identificar los distintos **alcances de variables**⁵⁹ (locales o globales) en diferentes lenguajes de programación y su aplicación en los lenguajes utilizados.

Ampliar la noción de dato compuesto para incluir datos que son **secuencias y conjuntos de datos**.

Utilizar **tipos estructurados de datos**, que representen **conjuntos y secuencias** de datos (por ejemplo, listas), y comandos y expresiones para recorrerlos (por ejemplo, `foreach`), modificarlos y acceder a sus elementos.

Utilizar herramientas de **repetición indexada** (por ejemplo, `foreach`, iteradores, etc.) para expresar las estrategias de recorrido de secuencias y listas.

Adquirir fluidez en la utilización de tipos estructurados de datos (registros, listas y sus combinaciones).

⁵⁵ El tipo de una expresión es la denominación del conjunto al que pertenece la información descrita por dicha expresión.

⁵⁹ El alcance de una variable es la parte del programa donde el nombre de la variable tiene sentido y puede ser utilizado para recordar el valor asignado a la misma.

registros se pueden acceder con observadores de campo —por ejemplo, *damePalo* para las cartas—).

Considerar los **tipos de datos** involucrados en los argumentos de un comando o expresión y en los parámetros de procedimientos y funciones para interpretar y solucionar errores de ejecución.

Anticipar qué tipo de dato es necesario en una expresión a partir de la naturaleza de un operador.

Distinguir entre **datos simples** (números, cadenas) y **datos compuestos** que modelen entidades individuales (como una carta o una persona), e identificar los componentes de un dato compuesto (por ejemplo, cartas —con número y palo—, personas —con nombre, apellido y otros datos—, etc.).

Utilizar **tipos estructurados de datos** como los **registros**⁵⁶ que modelen entidades mediante datos compuestos.

Reconocer la importancia de construir programas que sean entendibles por las personas, y utilizar las herramientas brindadas por el lenguaje de programación para conseguirlo (en particular, uso de procedimientos y funciones para la división en subtareas, definición de tipos de datos adecuados —incluyendo secuencias—, y denominación representativa de los identificadores).

⁵⁶ Los registros son una clase de tipos de datos de los lenguajes de programación cuyos elementos son datos estructurados que consisten en una colección fija de campos que describen varios otros datos más simples de diferentes tipos. Se puede acceder a los diferentes campos de un registro usando el nombre del campo sobre dicho registro.

		<p>Reconocer la importancia de construir programas que sean entendibles por las personas, y utilizar las herramientas brindadas por el lenguaje de programación para conseguirlo (en particular, uso de procedimientos y funciones para la división en subtareas, elección de tipos de datos adecuados y denominación representativa de los identificadores).</p>	
<p>4. Ejecución secuencial y paralela de programas.</p>	<p>Comprender que la ejecución de un programa consiste en que un autómatas lleve a cabo las acciones indicadas por cada instrucción.</p> <hr/> <p>Comprender que correr un programa secuencial consiste en la realización paso a paso de las acciones descritas por las instrucciones especificadas en el programa, y que la ejecución consiste en que la computadora efectivamente lleve a cabo dichas acciones.</p>	<p>Reconocer que existen modelos de ejecución no secuenciales como el concurrente o paralelo y aplicaciones que utilizan estos modelos (tales como los personajes de una animación que ejecutan código en forma concurrente, juegos MMORPGs —Massively Multiplayer Online Role Playing Games— donde los jugadores llevan diferentes procesos en paralelo, cajeros automáticos corriendo en paralelo sobre la misma base de información del banco, documentos colaborativos en línea).</p> <hr/> <p>Reconocer los eventos como la herramienta que permite la ejecución de</p>	<p>Comprender que las aplicaciones actuales (en la web o en los celulares) se basan en modelos de ejecución paralela en donde interactúan servicios y componentes de diversos orígenes.</p>

	<p>Reconocer que existen modelos de ejecución secuencial, interactiva y de varios procesos secuenciales que se ejecutan en forma simultánea.</p>	<p>un fragmento del programa asociada a alguna ocurrencia particular en el entorno (por ejemplo, se presiona una tecla, se recibe un mensaje, se cambia la orientación del celular).</p>	
<p>5. Compilación e interpretación: intermediación entre el código fuente y la ejecución.</p>	<p>Reconocer la interpretación de un programa como proceso en el cual un programa (el intérprete) toma cada una de las instrucciones y las lleva a cabo efectivamente.</p> <p>Reconocer los entornos de programación por bloques como programas que interpretan los programas creados por las y los usuarios de dicho entorno.</p>	<p>Construir la idea de código fuente como la entidad que se construye (en términos de bloques o texto) y se puede guardar en un archivo y compartir.</p> <p>Reconocer que para poder ejecutar un código fuente existe un proceso de intermediación que puede ser dado por un intérprete o un compilador propio de cada lenguaje de programación.</p> <p>Construir la idea de código ejecutable como aquel que puede ser ejecutado efectivamente por una máquina específica sin ningún proceso de intermediación (por ejemplo, el archivo <code>.exe</code>).</p>	<p>Comprender la diferencia entre el proceso de compilación (que dado un código fuente de un lenguaje produce un código fuente en otro lenguaje) y la ejecución de código en forma interpretada (donde hay un programa intermediario que va analizando y ejecutando cada instrucción de a una por vez).</p> <p>Comprender la necesidad de que exista al menos un programa escrito en código ejecutable de la máquina para cada tipo de máquina que se quiera utilizar (puede ser únicamente un intérprete, por ejemplo, la Máquina Virtual de Java), y que ese programa se puede escribir directamente en el lenguaje de la máquina, o a través de un proceso de compilación.</p>

		<p>Entender la compilación como la transformación del código fuente de un programa construido en un lenguaje en el código fuente de un programa construido en otro lenguaje, usualmente en términos de código ejecutable.</p>	
--	--	--	--

C. Área Infraestructura tecnológica

Las computadoras, tal como se las conoce hoy en día, se crearon con el propósito de procesar información siguiendo programas. Se interactúa con computadoras en forma creciente: los celulares inteligentes (smartphones), los dispositivos de control de acceso, los cajeros automáticos, las terminales de pago con tarjetas, entre otros, son ejemplos de equipamientos que son computadoras o contienen dispositivos computacionales.

Para poder evaluar y elegir los dispositivos computacionales que respondan a un determinado problema o necesidad, es necesario conocer cómo está construido. Para tener una mirada crítica de estos y la infraestructura asociada, se requiere, además, conocer los componentes y la arquitectura de las computadoras, identificar el rol de los sistemas operativos y comprender el funcionamiento de internet y de las redes.

Eje C1. Organización y arquitectura de computadoras

La comprensión del funcionamiento de los dispositivos computacionales con los que interactúan las personas a diario requiere conocer cuáles son sus componentes fundamentales, cómo están organizados y cómo se relacionan entre sí para ejecutar programas.

Esta visión de sistema no solo permite una comprensión más completa, sino que habilita a las y los estudiantes a diagnosticar problemas de mal funcionamiento e interpretar especificaciones técnicas para tomar un rol activo al momento de elegir las prestaciones de sus dispositivos o participar en comunidades especializadas.

En la siguiente tabla se describen los alcances previstos para cada uno de los ciclos escolares en relación a los temas que pertenecen a este eje.

Área →	Área Infraestructura tecnológica		
Eje →	Organización y arquitectura de computadoras		
Saberes →	Primaria - 2º Ciclo	Secundaria - Ciclo Básico	Secundaria - Ciclo Orientado
	<p>El reconocimiento de las computadoras como máquinas programables que involucran partes físicas (hardware) y partes lógicas (software) que interactúan entre sí para lograr el funcionamiento esperado. Y la identificación de las funciones de los principales componentes físicos involucrados, para comprender que todo sistema con este tipo de componentes es una forma de computadora, más allá de su forma física.</p>	<p>La comprensión del modelo de Von Neumann como una forma general de organizar una máquina programable, común a la gran mayoría de computadoras, y el reconocimiento del rol de los componentes fundamentales para reconstruir la ejecución de un programa hasta la ejecución de las instrucciones en el procesador.</p> <p>El conocimiento de las especificaciones técnicas de los componentes de hardware para comprender su relación con el desempeño de la computadora y poder elegir componentes en función de sus necesidades de desempeño.</p>	
Temas ↓	Alcances ↓		
	Primaria - 2º Ciclo	Secundaria - Ciclo Básico	Secundaria - Ciclo Orientado

1.

La computadora como un sistema integrado por hardware y software.

Comprender que el **hardware** incluye los componentes físicos de una computadora (como la memoria, pantallas, discos, etc.) y que el **software** son los programas que se ejecutan sobre dicho hardware.

Reconocer que una computadora es un sistema en el cual interactúan diferentes componentes de hardware y software que se comunican entre sí para que funcione de la forma esperada.

Reconocer que tanto el hardware como el software son imprescindibles para el funcionamiento de una computadora.

Reconocer que las **soluciones computacionales** pueden implementarse tanto mediante software como mediante hardware, o combinaciones de ambas, así como reconocer beneficios y limitaciones de cada una.

Reconocer que el software tiene necesidades específicas de hardware y saber anticiparlas.

Identificar cómo participan, se relacionan y comunican los diferentes componentes al realizar tareas específicas (qué pasa en el sistema al encenderse, al ejecutar un programa, cuando algo falla, etc.).

2.

Modelo de máquina programable: arquitectura von Neumann.

Reconocer que las computadoras no solo se encuentran en las PC tradicionales o los laptops, sino también en muchísimos dispositivos que forman parte de la vida contemporánea (por ejemplo, en teléfonos celulares, televisores, automóviles, lector de tarjeta de transporte, cajero automático, etc.).

Completar la noción de computadora con el modelo de arquitectura clásica de von Neumann.

Entender cómo los componentes de la arquitectura clásica de von Neumann participan en la ejecución de un programa (la CPU busca

	<p>Comprender que una computadora es una máquina programable que recibe información, la procesa (realiza operaciones, comparaciones, modificaciones, o la almacena) y genera nueva información (una acción, una imagen, un texto) sobre la base de un programa.</p>	<p>datos e instrucciones de programas en la memoria, los procesa y los vuelve a grabar en la memoria).</p> <p>Reconocer la vigencia de la arquitectura de von Neumann en los dispositivos computacionales que utilizan a diario.</p>	
<p>3. Componentes de hardware.</p>	<p>Conocer la función de los componentes fundamentales de una computadora: procesador, memoria RAM, dispositivos de almacenamiento y dispositivos de entrada y salida.</p> <p>Comprender que todos los datos digitales que conservamos están almacenados en algún dispositivo de almacenamiento (siempre existe un soporte físico, ya sea local o remoto).</p> <p>Conocer la existencia de diferentes unidades de medida para describir la</p>	<p>Vincular los componentes fundamentales de hardware con el rol que ocupan en el modelo de von Neumann y cómo se comunican entre sí.</p> <p>Conocer e interpretar las especificaciones técnicas básicas de los componentes fundamentales.</p> <p>Reconocer que el mayor tiempo de procesamiento está en leer y escribir en la memoria y en los discos y la función de la memoria caché para amortiguar ese problema.</p>	

capacidad de almacenamiento (KB, MB, GB).

Comprender que contar con un procesador que tiene varios **núcleos** habilita posibilidades de procesamiento en paralelo, lo cual permite mejorar la velocidad del sistema cuando realiza muchas tareas en simultáneo.

Distinguir diferentes escenarios de decisiones a la hora de **seleccionar una computadora** que involucren necesidades de usuario, costos, funcionalidades, velocidad, almacenamiento (por ejemplo, una computadora para juegos necesita contar con mayor capacidad de cómputo y memoria, por lo que tendrá un precio alto; por otra parte, una computadora para juegos con mucho poder de cómputo, pero poca memoria, implica un balanceo incorrecto y un comportamiento no esperado).

Incorporar **vocabulario específico** para poder comunicarse con profesionales (servicio técnico o sitios de venta), participar en foros dedicados al análisis de dispositivos computacionales y de solución de problemas y leer especificaciones o manuales técnicos.

4.

Ejecución a bajo nivel: lenguaje de máquina, representación binaria e impulsos eléctricos.

Comprender el **lenguaje de máquina** como un lenguaje capaz de ser interpretado directamente por el procesador, formado por un conjunto reducido de instrucciones simples.

Comprender que los programas creados en cualquier lenguaje de programación tienen que ser traducidos al lenguaje de máquina para poder ser ejecutados por la computadora, o deben poder ser interpretados por un programa que esté en ese lenguaje.

Comprender que el hardware se organiza basándose en **impulsos eléctricos**, y de ahí extrapolar la necesidad de representar la información con **dígitos binarios**.

Comprender que toda la información que se procesa y almacena en una computadora (por ejemplo, texto, imágenes o video) es representable a través de combinaciones de unos y ceros (que expresan los estados de presencia o ausencia de electricidad en un cable o magnetismo en diferentes componentes), y que en definitiva procesar dicha información consiste en un cómputo que realiza operaciones sobre estas representaciones.

		<p>Comprender cómo se relacionan las diferentes partes del procesador (contador de programa, ALU, unidad de control) y la memoria RAM al momento de ejecutar una instrucción del lenguaje de máquina (el ciclo de instrucción).</p> <p>Comprender que los diferentes componentes están contruidos por circuitos electrónicos que permiten operar y almacenar la información representada con electricidad.</p>	
--	--	---	--

Eje C2. Sistemas operativos

Los sistemas operativos (SO) están presentes en los dispositivos computacionales que utilizan las personas y cumplen una función central para organizar el uso compartido de los recursos de hardware que permite la ejecución de los programas. El sistema operativo brinda interfaces y abstracciones que permiten que usuarias y usuarios operen sobre software sin preocuparse por lo que necesita el hardware (por ejemplo, manipular archivos y no un dispositivo de almacenamiento). Conocer las funcionalidades e interfaces que ofrece el sistema operativo permitirá a las y los estudiantes identificar cómo impactan estas en la experiencia de uso de los dispositivos computacionales contemporáneos.

Existen diferentes sistemas operativos, cada uno ofrece un sistema de archivos, un conjunto de controladores y un modo de administrar los recursos que puede ser más eficiente o no.

En la siguiente tabla se describen los alcances previstos para cada uno de los ciclos escolares en relación a los temas que pertenecen a este eje.

Área →	Infraestructura tecnológica		
Eje →	Sistemas operativos		
Saberes →	Primaria - 2º Ciclo	Secundaria - Ciclo Básico	Secundaria - Ciclo Orientado
	<p>El reconocimiento del sistema operativo como el programa central presente en los dispositivos computacionales, encargado de organizar el funcionamiento y la interacción entre hardware y software.</p> <p>El reconocimiento de los archivos como la unidad de almacenamiento de información y su organización en directorios como una forma eficaz.</p>	<p>El reconocimiento del sistema operativo como administrador de recursos de hardware y como interfaz entre hardware y software para dimensionar su injerencia en el funcionamiento del sistema.</p> <p>El reconocimiento del rol del sistema de archivos para interactuar con los dispositivos de almacenamiento y el rol de los procesos para la ejecución de múltiples programas.</p>	<p>La construcción de una visión de capas e interfaces del sistema compuesta por el hardware, los controladores, el sistema operativo, las bibliotecas de sistema y el software de aplicación.</p> <p>El reconocimiento de la relevancia de las máquinas virtuales como un entorno para proteger y compartir recursos de hardware, en los servicios de internet de uso contemporáneo.</p>
Temas ↓	Alcances ↓		
<p>1.</p> <p>El sistema operativo como administrador de recursos e interfaz para el hardware.</p>	Primaria - 2º Ciclo	Secundaria - Ciclo Básico	Secundaria - Ciclo Orientado
	<p>Reconocer que los dispositivos computacionales que permiten usar varias aplicaciones (notebook, celular, PC, smart TV) tienen un sistema operativo (SO).</p>	<p>Reconocer al SO como una solución de software que administra los recursos de hardware entre los múltiples programas que se ejecutan en la computadora: planificación del uso del procesador, administración de memoria, de los dispositivos de entrada y salida, y de archivos, etc.</p>	<p>Reconocer que existen bibliotecas que se usan en el desarrollo de programas de usuario para acceder a las funcionalidades que el SO ofrece (por ejemplo, interactuar con un componente de hardware en</p>

Reconocer al SO como un **programa** central que **organiza** los componentes de hardware y software y al que recurren los programas para **interactuar con el hardware**.

Conocer que **existen diferentes sistemas operativos** (Unix, Windows, Android OS, IOS, etc.).

Reconocer que los programas que utilizamos habitualmente están contruidos para un sistema operativo en particular, y por ello es necesario atender con qué sistema operativo son **compatibles** los programas que queremos usar.

Reconocer que para que el sistema operativo pueda controlar los componentes de hardware existe un software llamado **controlador** (*driver* en inglés) que se debe instalar para cada dispositivo y debe ser compatible con el sistema operativo.

Comprender la importancia del sistema operativo como una **interfaz entre el hardware y el software** de manera que se pueda modificar un componente de hardware sin cambiar las aplicaciones de software, y viceversa.

Reconocer a los **controladores** como una interfaz entre el SO y el hardware que permite al SO interactuar con una variedad de dispositivos de hardware diferentes y que permite modificar el hardware sin tener que modificar el SO.

Comprender que un controlador es un componente de software que permite que el sistema operativo acceda a las funciones de un determinado hardware.

Reconocer que **cada sistema operativo hace un uso singular** de los recursos de hardware, que puede ser más o menos eficiente.

particular, reservar memoria, operar con archivos, etc.) y que estas bibliotecas constituyen una **interfaz** entre las aplicaciones y el SO.

2.

Abstracciones del sistema operativo: archivos y sistemas de archivos, procesos, virtualización y usuarios.

Reconocer que toda la información que conserva un dispositivo computacional se guarda en un **archivo** y todos los archivos se organizan en **directorios** y esto está reflejado en la **ruta** del archivo.

Reconocer que para poder explorar, recuperar, modificar y eliminar información de un dispositivo computacional es importante organizar de forma eficaz los **directorios** y utilizar **nombres** para los archivos y directorios que se relacionen con el contenido o la información que almacenan.

Reconocer que los archivos cuentan con **propiedades**: nombre, tipo, tamaño, autor, permisos (escritura, lectura, ejecución), fechas de creación y modificación.

Reconocer que la **extensión de archivo** es parte del nombre, que nos permite declarar el tipo del archivo, y que es información que el sistema operativo puede utilizar para elegir un programa para abrirlo.

Reconocer que el SO brinda herramientas al usuario administrador para generar otros

Reconocer la importancia del **sistema de archivos** en tanto permite interactuar con los dispositivos de almacenamiento viéndolos como **particiones** con directorios y archivos.

Diferenciar dispositivo de almacenamiento, partición y sistema de archivos.

Reconocer que existen **diferentes sistemas de archivos**, y por defecto cada sistema operativo es compatible con algunos sistemas de archivos.

Reconocer que todo lo que se ejecuta en una computadora es parte de un **proceso**.

Reconocer que el comportamiento que vemos de un programa suele involucrar la ejecución de múltiples procesos.

Comprender que el SO se ocupa de administrar los recursos (espacio de memoria, tiempo de procesador, acceso a la

Reconocer que una **máquina virtual** es una simulación de una computadora, y por lo tanto los efectos sobre ella no impactan sobre el resto del sistema.

Reconocer las máquinas virtuales como un entorno de software que permite ejecutar sistemas operativos.

Reconocer la existencia de las máquinas virtuales como un entorno para proteger y compartir recursos de hardware, aislando por completo los programas que corren en las máquinas virtuales entre sí y de la máquina anfitriona (por ejemplo, en los servidores que soportan múltiples servicios o usuarios).

	usuarios con diferentes permisos y limitaciones de acceso a archivos y programas.	red, etc.) que necesitan los procesos para ser ejecutados de forma simultánea.	
--	--	--	--

Eje C3. Redes e internet

Con frecuencia, las personas utilizan internet en la vida cotidiana sin cuestionarse su funcionamiento y estructura. Por ello, este eje propone desarmar y examinar detalladamente la red de internet para comprender cómo funciona y cómo las personas se involucran en ella. Se abordan tanto aspectos técnicos como políticos de esta red mundial de circulación de datos, desde sus componentes físicos fundamentales hasta sus diferentes capas y protocolos.

En la siguiente tabla se describen los alcances previstos para cada uno de los ciclos escolares en relación a los temas que pertenecen a este eje.

Área →	Infraestructura tecnológica		
Eje →	Redes e internet		
Saberes →	Primaria - 2º Ciclo	Secundaria - Ciclo Básico	Secundaria - Ciclo Orientado
	La comprensión de internet como una red de computadoras que intercambian información entre sí, sus componentes físicos fundamentales, su organización y el recorrido de la información, para reconocer la necesidad de mecanismos de seguridad para un intercambio seguro y de un servidor que permita a las aplicaciones de internet	La comprensión de la red internet como un modelo de capas donde cada capa resuelve un problema y brinda una funcionalidad a la capa siguiente asociada a un protocolo. El reconocimiento de diferentes tipos de modelos que permiten distribuir tareas y trabajo a los diferentes dispositivos conectados (modelos cliente-servidor y P2P), sus ventajas y limitaciones.	El reconocimiento de la existencia de aplicaciones que permiten controlar el flujo y la anonimización en el intercambio de información.

	acceder, almacenar y procesar la información.	La utilización de herramientas de diagnóstico para identificar problemas y conocer el estado de la red.	
Temas ↓	Alcances ↓		
	Primaria - 2º Ciclo	Secundaria - Ciclo Básico	Secundaria - Ciclo Orientado
1. Estructura y funcionamiento de redes de computadoras. Internet como una red.	<p>Conocer las redes de computadoras como un conjunto de computadoras interconectadas entre sí de manera que pueden intercambiarse información entre ellas.</p> <hr/> <p>Reconocer distintas redes de computadoras que forman parte de la vida contemporánea (por ejemplo, redes domésticas o escolares), algunos de sus componentes fundamentales (routers y hosts), su organización y el recorrido de la información.</p> <hr/> <p>Identificar internet como la red de computadoras que se comunican entre sí intercambiando información a escala mundial y con millones de usuarios.</p>	<p>Comprender que el diseño de internet es un modelo de capas donde cada una resuelve un problema específico y brinda una funcionalidad a la capa siguiente.</p> <ul style="list-style-type: none"> Reconocer que en cualquier comunicación existe un medio de transmisión cableado (cable de cobre, fibra óptica) o un medio inalámbrico (ondas de radio) y que es utilizado directamente por la capa física. Reconocer los enlaces de datos como una manera de intercambiar información entre un conjunto acotado de computadoras, que permite identificar pérdidas de información que pueden haber sucedido en la transmisión de la capa física. Reconocer que la capa de red agrega la posibilidad de intercambiar información entre un conjunto de computadoras de mucho mayor escala (en distancia y en cantidad). Reconocer que la capa de transporte permite transmitir de manera confiable cantidades arbitrarias de información entre aplicaciones resolviendo pérdidas y alteración del orden de la 	

Reconocer los **enlaces** que forman una red y las tecnologías más usuales (ethernet, wifi, 4g).

Reconocer los **enlaces físicos** sobre los que se construye internet a nivel mundial, como los cables de enlace submarino, enlaces satelitales, etc.

Reconocer que el envío de información a través de una red requiere del paso por **puntos intermedios**.

Conocer que al enviar información a través de internet, se traslada por medios físicos, se fragmenta en **paquetes** para su traslado, se recompone al llegar a destino y que cada paquete puede seguir una **ruta** diferente.

Identificar si la imposibilidad de conectarse a internet corresponde a un problema del servicio o algún

información propias de las limitaciones de la capa de red.

- Reconocer que la **capa de aplicación** contiene a los programas que utilizan la red para el intercambio de información.
-

Comprender la noción de **protocolo** de red como un conjunto de reglas de cómo deben comportarse los dispositivos y artefactos individuales para garantizar un comportamiento particular del conjunto.

Asociar las funciones de cada capa del modelo con la existencia de un protocolo específico:

- Identificar que **Ethernet** y **wifi** son protocolos de la **capa de enlace** y, por lo tanto, son un conjunto de reglas que permiten utilizar medios físicos para intercambiar información entre un conjunto acotado de computadoras.
- Identificar que para que la **capa de red** funcione es necesario identificar las computadoras y retransmitir los paquetes de manera que lleguen a destino (**enrutamiento**) y que este problema se resuelve de manera distribuida y dinámica mediante el **protocolo IP**.
- Identificar que para que la **capa de transporte** cumpla su función se fragmenta la información en paquetes numerados del mismo tamaño y se define un **puerto** como un identificador que permite reconocer a qué programa debe entregarse la información; y que este problema

problema de los dispositivos de la red local.

se resuelve mediante protocolos de transporte, como **TCP** o **UDP**.

- Conocer que los programas de la **capa de aplicación** definen protocolos propios para su funcionamiento (por ejemplo, **HTTP** para páginas web y **BitTorrent** para el intercambio de archivos) que funcionan sobre los protocolos de transporte.

Valorar los **protocolos abiertos** porque permiten construir distintos softwares para las aplicaciones (por ejemplo, los protocolos abiertos de la web permiten que se construyan una variedad tanto de navegadores como servidores web, a diferencia de las aplicaciones de mensajería privativas, donde solo podemos usar la aplicación que provee la empresa dueña de la plataforma).

Reconocer que dentro de una red privada los dispositivos se identifican con direcciones **IP privadas** y que al acceder a internet, todos lo hacen con la misma **IP pública**.

Reconozcan que las **URL** incluyen una referencia a una computadora de la red mediante su identificación IP y que la traducción entre nombres de dominio y direcciones IP es realizada por los servicios de **DNS**.

		<p>Conocer las métricas más habituales para medir eficiencia en internet (ancho de banda, latencia, etc.) siendo capaces de interpretar en forma práctica algunos valores de tales medidas (por ejemplo, la diferencia entre una conexión de 100 Mbps y una velocidad de bajada de 20 Mbps).</p> <hr/> <p>Utilizar herramientas de diagnóstico de redes (por ejemplo, ping, traceroute) para conocer el estado de la red, identificar problemas y proponer soluciones.</p> <hr/> <p>Identificar que los grandes puntos de interconexión y los grandes enlaces pertenecen a un grupo reducido de empresas asociadas a países centrales y cómo esto impacta en la provisión del servicio y la neutralidad de la red.</p>	
<p>2.</p> <p>Aplicaciones en internet: almacenamiento o remoto, www, modelo cliente-servidor y P2P.</p>	<p>Comprender que las aplicaciones que funcionan en internet están conformadas por diversos programas que funcionan en diferentes computadoras intercambiando información a través de la red.</p> <hr/>	<p>Conceptualizar el modelo cliente-servidor como una forma de organizar las tareas de las aplicaciones utilizadas en internet.</p> <hr/> <p>Reconocer los roles de cliente y servidor en relación a las tareas que realiza cada uno para el funcionamiento de una aplicación (por ejemplo, en una aplicación de</p>	

Reconocer que el funcionamiento de ciertas aplicaciones en red (en particular internet) necesitan de un **servidor**, es decir, de una computadora que esté disponible todo el tiempo, para poder acceder, almacenar y procesar información.

Distinguir qué información reside en su dispositivo de cómputo (computadora, celular, consola, etc.) y cuál en servidores o en otros dispositivos.

Reflexionar sobre la metáfora de “**la nube**” y su concreción a través de servidores que se encuentran en grandes centros de datos.

Reconocer que la información y los datos que se suben o comparten en internet (redes sociales, aplicaciones, videojuegos online, etc.) son almacenados en **servidores controlados por las empresas dueñas de los servicios utilizados**.

mapas, el cliente brinda una interfaz para elegir un lugar, y el servidor se encarga de calcular y enviar la ruta).

Conocer el **modelo P2P** como una manera de organizar una aplicación en la que todas las computadoras pueden interactuar entre sí sin roles diferenciados.

Contrastar las **posibilidades de lucro y de control** (por ejemplo, del tráfico de datos, del servicio, de usuarios) entre el modelo cliente-servidor y el modelo P2P, en cuanto en el primero el control es unilateral y en el segundo el control se encuentra distribuido y es por lo tanto más complejo de direccionar.

Identificar que la existencia de las aplicaciones actuales de internet depende de **servidores** y grandes **centros de cómputos** que son propiedades de un **grupo reducido de empresas**.

Reconocer distintas aplicaciones (un navegador, un cliente de mensajería, una aplicación de videoconferencia, una aplicación de contenidos multimedia) que **utilizan internet como un canal de comunicación.**

Reconocer los navegadores, repositorios remotos de almacenamiento, servicios web, entre otros, como distintas categorías de las funciones de las aplicaciones de internet (Firefox, Chrome, Chromium, Opera, Safari cumplen la función de navegación).

Identificar la **web** (World Wide Web o simplificado www) como un sistema particular en el que un navegador (Firefox, Chromium, etc.) intercambia información a través de internet con un servidor y la visualiza en formato de páginas web.

3.

Mecanismos de seguridad en redes: protocolos, criptografía y anonimización.

Comprender que el uso de redes expone al usuario a diferentes ataques o problemas de seguridad porque la computadora puede ser accesible desde cualquier otra computadora de la red y la información enviada puede ser interceptada por las computadoras intermedias.

Conocer la necesidad de la **encriptación** como forma de proveer seguridad en el intercambio de información en redes y reconocer que está presente en muchas aplicaciones.

Conocer que a los usuarios de una red se le pueden asignar diferentes **permisos** (por ejemplo, solo acceder a internet y no al resto de los dispositivos de la red).

Conocer que existen **protocolos seguros para la capa de transporte** y que, por lo tanto, pueden brindarle seguridad a las aplicaciones de la capa de aplicación sin tener que modificarlas (al agregar SSL/TLS sobre TCP, el protocolo HTTP se convierte en un protocolo seguro, que se suele llamar **HTTPS**).

Reconocer que los protocolos seguros de transporte (SSL/TLS) utilizan **cadena de certificados** que permiten autenticar al sitio y encriptar la comunicación.

Tomar conciencia de que al utilizar internet siempre figura la dirección IP del usuario y que esto puede ser utilizado para identificarlo y geolocalizarlo, y que además, las aplicaciones pueden enviar información extra (por ejemplo, los navegadores envían información sobre el sistema operativo, modelo del navegador, tamaño de la pantalla, detalles de hardware, etc.) que también puede usarse para **identificar a un usuario**.

Conocer las **cookies** como una manera en que los sitios web almacenan información en nuestros dispositivos (por ejemplo, sesiones iniciadas o preferencias de búsqueda) y cómo pueden ser utilizadas para rastrear información de las acciones de un usuario en distintas páginas de internet.

Conceptualizar las aplicaciones de VPN como una simulación por software de una red privada entre computadoras conectadas a una red pública.

Conocer que existen herramientas que intentan **anonimizar el uso de internet** (por ejemplo, TOR).

Conocer que existen herramientas que permiten controlar el tráfico entrante y saliente tanto de un dispositivo como de una red (por ejemplo, Firewall, cortafuegos).

		<p>Conocer la existencia de redes privadas como un mecanismo de seguridad que permite limitar quiénes participan de la red y vincularlo con el requerimiento de contraseña para conectarse a la red.</p>	
--	--	---	--

D. Datos e inteligencia artificial

Los datos son recolectados, almacenados, analizados y utilizados para sacar conclusiones y hacer predicciones sobre comportamientos futuros. Pero los datos recolectados siempre corresponden al pasado, su registro es imperfecto y sesgado, ya que dependen de los valores, creencias e ideas de quienes desarrollan las herramientas para recolectarlos.

El objetivo de esta área es conocer la recolección, representación y procesamiento de datos. Además, se abordan los procesos de análisis, consulta y transformación de los datos para realizar conclusiones. En esta misma línea, se abordan contenidos de inteligencia artificial y aprendizaje automático, prestando especial atención a sus limitaciones, la subjetividad de las personas y los datos involucrados en su desarrollo y las consecuencias de su uso.

Eje D1. Recolección y modelado de datos

El funcionamiento de los dispositivos y artefactos computacionales contemporáneos suele involucrar la recolección de datos, a veces de manera más explícita que otras. Reconocer cuándo esto sucede y comprender los mecanismos que lo posibilitan es necesario para construir una mirada crítica y precisa sobre las experiencias de las personas, y da sustento a las reflexiones y preocupaciones que se abordan en el eje Computación y sociedad. Además, poner en práctica el modelado y la recolección de datos de forma crítica será el primer paso para que las y los estudiantes analicen situaciones o fenómenos con herramientas basadas en datos.

En la siguiente tabla se describen los alcances previstos para cada uno de los ciclos escolares en relación a los temas que pertenecen a este eje.

Área →	Datos e inteligencia artificial		
Eje →	Recolección y modelado de datos		
Saberes →	Primaria - 2º Ciclo	Secundaria - Ciclo Básico	Secundaria - Ciclo Orientado
	El reconocimiento y la experiencia de situaciones que involucran recolección de datos en forma manual y automatizada.	El modelado de entidades con atributos para recolectar datos en manera activa que se almacenan en planillas, entendidas como forma elemental de base de datos.	La comprensión de las nociones básicas de base de datos relacionales para almacenar y recuperar datos que permitan sacar conclusiones. La valoración de los conjuntos de datos abiertos como una manera de compartir conocimiento.
Temas ↓	Alcances ↓		
	Primaria - 2º Ciclo	Secundaria - Ciclo Básico	Secundaria - Ciclo Orientado
1. Recolección de datos activa y pasiva.	<p>Reconocer diferentes formas de recolectar datos, que pueden o no involucrar dispositivos computacionales.</p> <hr/> <p>Conocer herramientas automatizadas de recolección de datos como encuestas en línea.</p> <hr/>	<p>Reconocer de qué manera los artefactos computacionales recolectan información, diferenciando cuándo recolectan la información de modo activo o pasivo (por ejemplo, solicitar información mediante un formulario o un programa que registre la información de sitios visitados).</p> <hr/> <p>Diseñar encuestas en línea utilizando diferentes tipos de preguntas para recolectar datos en función de los modelos</p>	<p>Reconocer la existencia de conjuntos de datos de acceso abierto recolectados por diferentes organizaciones y valorarlos como una forma de compartir conocimiento.</p> <hr/> <p>Identificar si existen conjuntos de datos de acceso abierto que pueden ser utilizados para analizar un determinado fenómeno.</p> <hr/>

	<p>Reconocer en el funcionamiento de los artefactos computacionales contemporáneos la recolección de datos, en qué forma sucede y para qué son utilizados.</p>	<p>que definan para resolver una situación problemática.</p> <hr/> <p>Reflexionar sobre la importancia de conocer cuándo se está realizando recolección pasiva de datos y sobre el hecho de que no está permitido por ley recolectar datos personales sin aviso.</p> <hr/> <p>Reconocer que la recolección de datos necesita de un almacenamiento físico y dimensionar su escala en aplicaciones contemporáneas.</p> <hr/> <p>Reconocer que es posible obtener una variedad de datos de la mayoría de fenómenos u objetos y que para esto son necesarias herramientas de recolección.</p>	<p>Reconocer si es suficiente contar con una herramienta digital de recolección activa o se necesita contar con un programa que automatice la recolección de datos de forma pasiva.</p>
<p>2. Modelado de datos: entidades con atributos. Base de datos.</p>	<p>Identificar en una planilla de datos de una encuesta cómo se registra cada caso (por ejemplo, una fila) y la información asociada a ese caso</p>	<p>Reconocer que existen diferentes características (atributos) que nos permiten modelar diferentes elementos (entidades) para recolectar información relevante y analizarla.</p>	<p>Identificar los motores de bases de datos como la pieza de software que implementa bases de datos de forma eficiente, garantizando ciertas propiedades fundamentales de los datos (por ejemplo, consistencia, confidencialidad, acceso</p>

(por ejemplo, una columna por pregunta).

Identificar entidades y atributos en situaciones contemporáneas en las que haya recolección de datos.

Proponer un **modelo de entidades con atributos** para recolectar datos.

Organizar los datos de **entidades con atributos en una planilla de cálculo** propia (por ejemplo, utilizando una fila para cada entidad y las columnas para cada atributo) e interpretar en estos términos una planilla con este tipo de datos generados por otras personas.

Aproximarse a la noción de **base de datos** a partir de reconocer en situaciones contemporáneas la presencia de conjuntos de datos almacenados y organizados cuyo procesamiento o registro son centrales para el funcionamiento de un artefacto computacional (por ejemplo, cuando se realiza una compra en un supermercado se consulta a la base para asociar un código con un precio, cuando se pide un libro en una biblioteca se utiliza la base para dejar registro de quién pidió qué libro y cuándo lo tiene que devolver, cuando se busca en un

simultáneo de muchos usuarios), y que permite ejecutar las consultas y verificar permisos.

Conocer los motores de bases de datos más frecuentemente usados, en especial, los de código abierto (MySQL).

Utilizar un **lenguaje de consultas** para recuperar información de una base de datos relacional aplicando proyecciones y filtros a partir de una o más tablas.

Comprender las nociones de **tabla, fila y columna** en una base de datos relacional en términos de **modelado de entidades con atributos**.

Comprender la noción de **clave primaria** como un atributo (o grupo de atributos) que permite identificar de manera única a una entidad en una base de datos (por ejemplo, el DNI o el CUIL en el caso de las personas) y su importancia para la recuperación de la información sobre esa entidad.

Conocer los **índices** de una tabla como una estructura de datos auxiliar que permite la

		<p>catálogo digital se consulta la base para seleccionar los productos que coinciden con la búsqueda y mostrarlos).</p> <hr/> <p>Reconocer que las planillas se pueden utilizar para implementar una forma elemental de base de datos.</p>	<p>recuperación de información de manera mucho más eficiente.</p> <hr/> <p>Diseñar una base de datos relacional e implementarla en un motor de base de datos, para almacenar datos y recuperarlos para sacar conclusiones.</p>
--	--	--	---

Eje D2. Ciencia de datos, inteligencia artificial y aprendizaje automático

Por un lado, la abundancia de datos registrados y el poder de cómputo ha convertido a las aplicaciones basadas en datos en actores cada vez más importantes para la ciudadanía. Por otro lado, los registros de datos son imperfectos y sesgados, ya que dependen de los valores, creencias, contextos sociohistóricos e ideas de quienes desarrollan las herramientas para recolectarlos.

En un mundo donde la injerencia de las ciencias de datos es cada vez más amplia, cobra relevancia dotar a las y los estudiantes de herramientas que les permitan analizar y procesar datos para así poder sacar sus propias conclusiones y evaluar de forma crítica las conclusiones que otros publican.

En este mismo escenario, resulta clave comprender cómo se generan las aplicaciones desarrolladas con IA, prestando especial atención a sus limitaciones, la subjetividad de las personas y los datos involucrados en su desarrollo y las consecuencias de su uso.

En la siguiente tabla se describen los alcances previstos para cada uno de los ciclos escolares en relación a los temas que pertenecen a este eje.

Área →	Datos e inteligencia artificial
--------	---------------------------------

Eje →	Ciencia de datos, inteligencia artificial y aprendizaje automático		
Saberes →	Primaria - 2º Ciclo	Secundaria - Ciclo Básico	Secundaria - Ciclo Orientado
	<p>El reconocimiento de la utilización de los datos para sacar conclusiones (a través de su visualización) y para generar modelos de aprendizaje automático que resuelven automáticamente problemas de clasificación.</p>	<p>La aproximación a la noción de inteligencia artificial como un conjunto de técnicas para desarrollar artefactos computacionales que no son inteligentes, sino que reproducen comportamientos humanos.</p> <p>El desarrollo crítico de modelos de aprendizaje automático que les permita profundizar los conceptos importantes en el proceso de desarrollo (conjunto de entrenamiento, entrenamiento, conjunto y métricas de evaluación, representatividad de los datos) y reflexionar sobre su impacto, teniendo en cuenta que estos artefactos reflejan subjetividades que pueden reproducir y profundizar desigualdades y prejuicios.</p> <p>La aplicación de herramientas de procesamiento automático de datos (filtros, transformaciones y totalizaciones) para generar visualizaciones y analizar un conjunto de datos.</p>	<p>La aplicación de herramientas de procesamiento y análisis automático de datos tanto para la preparación de conjuntos de datos como para la evaluación de su representatividad antes de usarlos para extraer conclusiones o entrenar y evaluar modelos de aprendizaje automático.</p> <p>El reconocimiento de las aplicaciones de inteligencia artificial como artefactos computacionales falibles, para los que en general no es posible explicitar el criterio que define y guía su funcionamiento.</p>
Temas ↓	Alcances ↓		
	Primaria - 2º Ciclo	Secundaria - Ciclo Básico	Secundaria - Ciclo Orientado
<p>1.</p> <p>Aplicaciones de inteligencia artificial. Mecanismos generales.</p>		<p>Reconocer que la inteligencia artificial (IA) es una denominación para referirse a artefactos computacionales que buscan imitar comportamientos humanos a distintos niveles (por ejemplo, un predictor de texto vs. un modelo de lenguaje generativo) y que pueden estar contruidos</p>	<p>Formar criterio para evaluar las respuestas que brindan las aplicaciones basadas en IA, contemplando desde una mirada crítica diversas variables (el conjunto de datos, el</p>

con técnicas muy diversas (por ejemplo, sistemas de reglas vs. redes neuronales pre-entrenadas).

Conocer que la mayoría de las aplicaciones “de IA” contemporáneas dependen de enormes volúmenes de datos y están construidas con **técnicas de aprendizaje automático (AA)**.

Reconocer que la IA tiene diferente grado de **aplicabilidad** según el tipo de problemas, y en qué tipo de aplicaciones y problemas es más frecuente encontrar soluciones exitosas basadas en modelos de aprendizaje automático (reconocimiento de imágenes, reconocimiento de voz o de imágenes, generación o traducción automática de texto, etc.).

Comprender que el **entrenamiento de un modelo** o aprendizaje de máquina consiste en encontrar patrones estadísticos de forma automática y que, por lo tanto, no constituye una forma de aprendizaje como lo hacemos los humanos, basada en el razonamiento.

Reconocer que el **comportamiento que se percibe como inteligente** en las aplicaciones de IA existe gracias a la inteligencia de los humanos (ya sea porque los humanos definen las relaciones lógicas o

nivel de representación de una población, país, región).

Reconocer que las aplicaciones basadas en IA cometen **errores** y que su éxito se mide en métricas estadísticas personalizadas para cada solución, que no se encuentran consensuadas y que generalmente no contemplan el impacto ético.

		<p>porque están basadas en ejemplos de humanos resolviendo el problema).</p> <hr/> <p>Reconocer que las aplicaciones de AA imitan el comportamiento humano que infieren a partir de un conjunto enorme de ejemplos y diferenciar esto de la forma en la que las personas resuelven un problema, utilizando su intuición, experiencia, conocimientos, razonamiento, creatividad, etc.</p>	
<p>2. Preparación, análisis y visualización de datos.</p>	<p>Identificar que la visualización de datos es una herramienta tanto para la comunicación como para el análisis de datos.</p> <hr/> <p>Crear gráficos estándares (de barra, de tortas, de líneas, etc.) para comunicar o analizar frecuencias de elementos de un conjunto de datos utilizando las herramientas de las planillas de cálculo.</p> <hr/> <p>Reconocer que un mismo conjunto de datos puede ser</p>	<p>Reconocer que para extraer conclusiones de un conjunto de datos es necesario procesarlo y visualizarlo.</p> <hr/> <p>Comprender que las operaciones de filtrado permiten seleccionar, del conjunto de datos, aquellos elementos que cumplen con una condición definida.</p> <hr/> <p>Comprender que las operaciones de transformación producen nuevos datos a partir de aplicar una transformación o una función a cada elemento de un conjunto de datos.</p>	<p>Reconocer la necesidad de la preparación de los datos como un primer paso de procesamiento (limpiar, recodificar, reordenar utilizando filtros, transformaciones y totalizaciones) para poder analizarlos.</p> <hr/> <p>Reparar problemas en un conjunto de datos (campos vacíos, erróneos, etc.) y decidir cómo solucionarlos (si es posible completarlos, reponerlos o eliminarlos).</p> <hr/> <p>Identificar durante el análisis de un conjunto de datos, datos atípicos o aislados (outliers) producto de errores de medición, así como</p>

presentado de diferentes formas para **resaltar distintos aspectos** (por ejemplo, una tabla de resultados se puede ordenar según la cantidad de derrotas, victorias o puntos, dependiendo de lo que se considere relevante).

Comprender que las **operaciones de totalización** calculan un valor a partir de un conjunto de elementos (por ejemplo la sumatoria, el promedio, la cantidad, el máximo, reordenamientos, etc.).

Analizar conjuntos de datos utilizando combinaciones de estas herramientas para responder preguntas y extraer conclusiones.

Formar criterio para elegir el **tipo de gráfico** más adecuado a los propósitos de visualización y comunicación que se persiguen.

Reconocer que las decisiones que se toman durante el procesamiento de un conjunto, así como las conclusiones que se obtienen de su análisis, dependen de interpretaciones subjetivas de quienes lo analizan.

Reconocer que el diseño de las herramientas de recolección es importante para no generar errores en los datos recolectados (campos vacíos o errores de tipo de dato).

su impacto en los resultados y la necesidad de filtrarlos.

Reconocer que las conclusiones a partir de la visualización y el análisis de datos no solo están afectadas por la subjetividad de quien interpreta los datos sino también por la representatividad de la muestra del conjunto de datos.

3.

Modelos de aprendizaje automático.

Reconocer aplicaciones contemporáneas cuyo funcionamiento depende de un conjunto de datos y, por lo tanto, pueden estar implementadas con aprendizaje automático (AA).

Crear modelos de aprendizaje automático de clasificación utilizando entornos diseñados para la enseñanza de IA:

- Analizar el problema para definir las clases.
 - Seleccionar y clasificar conjuntos de datos (imágenes, sonidos, etc).
 - Elegir datos para evaluar si el modelo responde a lo esperado.
-

Reconocer que las aplicaciones basadas en AA funcionan a partir de **reconocer y replicar patrones** detectados automáticamente en enormes volúmenes de datos.

Conceptualizar la **noción de clasificador** como un modelo de AA que toma un elemento del universo y le asigna una clase o categoría (seleccionada de entre un conjunto definido previamente por una persona).

Identificar la existencia de clasificadores y sus clases en aplicaciones de AA contemporáneas.

Comprender que un **modelo entrenado** de AA es un programa que se obtiene como resultado de un proceso automático de análisis de datos (entrenamiento).

Reconocer que es posible **generar modelos** para resolver problemas distintos utilizando el mismo entrenamiento y diferentes conjuntos de datos.

Identificar que un **modelo sin entrenar** es un programa que tiene valores no definidos y que el objetivo del entrenamiento es analizar un conjunto de datos para definir estos valores y que esto se hace mediante un programa.

Reconocer que el entrenamiento de un modelo de AA infiere **patrones en los datos** que luego

Conceptualizar el **entrenamiento** de un modelo de AA como la solución a un problema matemático en el cual se buscan valores de parámetros para que una función de clasificación minimice su error sobre un conjunto de datos dados.

Reconocer que en general no es posible comprender a partir de analizar el código fuente de un modelo entrenado qué criterio utiliza para clasificar o predecir (**explicabilidad**) a diferencia de los programas de código abierto desarrollados enteramente por personas.

Comprender que la significatividad y precisión de las salidas de un modelo dependen de la cantidad, calidad y diversidad de los datos recolectados.

Conocer otros algoritmos de ciencia de datos que permiten clasificar y **predecir sin necesidad de entrenar** un modelo de IA (por ejemplo, clustering, KNN).

Preparar de forma automática un conjunto de datos para entrenar un modelo de AA,

Reconocer que el desarrollo de aplicaciones con aprendizaje automático a nivel industrial necesita **enormes volúmenes de datos** y de **muchísimo poder de cómputo**.

reproducirá el modelo entrenado al funcionar, pero que el funcionamiento obtenido no necesariamente refleja las intenciones de los desarrolladores.

Reconocer que para inferir el comportamiento efectivo de un modelo es necesario **ponerlo a prueba sobre datos nuevos**.

Reconocer que al diseñar un clasificador basado en AA para resolver un problema, tanto la acción de definir clases como la de clasificar elementos implica emitir un **juicio sobre el problema**.

Dimensionar la **cantidad de datos y de energía** que se requieren para entrenar estos modelos y vincularlo con el eje de Computación y ambiente y el eje de Redes e internet.

Generar críticamente un modelo de AA para resolver un problema:

- Anticipar impactos del modelo teniendo en cuenta el contexto de uso.
- Definir las clases o categorías a partir de analizar el problema.
- Conseguir datos, teniendo en cuenta las categorías definidas y la representatividad de la muestra.

analizando posibles sesgos que puedan afectar la representatividad de la muestra.

Evaluar de forma automática un modelo de AA teniendo en cuenta distintos tipos de **métricas** y su interpretación (exactitud o *accuracy*, exhaustividad o *recall*, precisión o *precision*, F1, matriz de confusión) sobre un conjunto de datos de evaluación de volumen, teniendo en cuenta que no hay una de ellas que por sí sola garantice el buen funcionamiento.

Proponer ejemplos para poner a prueba modelos de AA que puedan explicitar sus sesgos.

		<ul style="list-style-type: none"> • Definir el conjunto de entrenamiento y de evaluación teniendo en cuenta su representatividad. • Poner a prueba el modelo sobre los datos de evaluación y tomar alguna métrica de desempeño (por ejemplo, exactitud o <i>accuracy</i>) teniendo en cuenta sus limitaciones. • Evaluar el impacto del modelo funcionando en su contexto de uso. 	
<p>4.</p> <p>Subjetividad expresada en los modelos. Sesgo algorítmico.</p>		<p>Comprender que las aplicaciones de AA son artefactos creados por personas en base a conjuntos de datos recolectados por personas y, por lo tanto, están sujetos a las subjetividades de estas personas.</p> <hr/> <p>Comprender que todo conjunto de datos refleja un recorte subjetivo de la realidad, tanto en el diseño (qué se recolecta) cómo en la recolección (de qué manera y quiénes participan de la muestra).</p> <hr/> <p>Comprender que los conjuntos de datos, en tanto son una muestra del mundo en un momento determinado, pueden reflejar las desigualdades existentes y, en ese caso, si son usados para</p>	<p>Comprender que un modelo de AA solo tendrá en cuenta aquello que figure en el conjunto de datos de entrenamiento y, por lo tanto, es fundamental que ese conjunto sea representativo del fenómeno que se quiere modelar y de la manera en la que se lo quiere modelar.</p> <hr/> <p>Comprender que las aplicaciones basadas en AA extraen generalizaciones de los conjuntos de datos a partir de encontrar regularidades y, por lo tanto, los subconjuntos de datos que reflejen situaciones poco representadas en el conjunto serán menos tenidos en cuenta.</p>

entrenar modelos el comportamiento de esos modelos puede **repetir y reforzar** esas desigualdades.

Aproximarse a la noción de **representatividad estadística** de una muestra para identificar sesgos en conjuntos de datos.

Aproximarse a la noción de **sesgo algorítmico** como patrones de comportamiento de un artefacto computacional, y saber que puede estar introducido tanto en el diseño del programa como por los datos de entrenamiento.

Reconocer que las desigualdades que dan origen al sesgo algorítmico de un sistema de AA se refuerzan a partir de la **retroalimentación**.

Diferenciar datos atípicos o *outliers* de datos correctos sobre poblaciones poco representadas.

Encontrar relaciones entre la subjetividad que expresa un conjunto de datos con **cómo, para qué y por quiénes** los conjuntos de datos son organizados, recolectados, transformados y etiquetados.

Reconocer que la **subrepresentación** en los conjuntos de datos suele estar asociada a **relaciones de poder**.

Prácticas computacionales

Se destacan cuatro prácticas computacionales por ser relevantes para acercar a las y los estudiantes a los “modos de hacer” computación. Se propone incorporarlas en situaciones de enseñanza habituales. Es en el marco de estas prácticas que los conceptos incorporados en esta propuesta curricular, se construyen, adquieren sentido y se organizan. Las prácticas son las siguientes.

- A. Reconocer y desarrollar soluciones para problemas computacionales.
- B. Crear y/o modificar artefactos computacionales.
- C. Desarrollar, utilizar y abrir abstracciones.
- D. Identificar y comunicar problemas de dispositivos y artefactos computacionales.

A. Reconocer y desarrollar soluciones para problemas computacionales

La utilización de la tecnología de la computación para la resolución de problemas se limita a determinados problemas con una serie de características específicas que denominamos *problemas computacionales*. Esta práctica involucra:

- identificar qué tipo de situaciones problemáticas pueden ser abordadas como problemas computacionales;
- conocer de forma integral los sistemas y las computadoras para definir y desarrollar una solución computacional, y
- dimensionar el impacto de la solución, tanto de las posibilidades que habilita como los nuevos problemas que puede generar.

B. Crear y/o modificar artefactos computacionales

Incentivar a las y los estudiantes a crear sus propios artefactos computacionales vinculados con sus intereses o que resuelvan problemas de su entorno potencia la apropiación de conceptos clave de la disciplina y contribuye a desarrollar la creatividad y la exploración de ideas. Se pueden crear [artefactos](#) o [dispositivos](#) (animaciones, videojuegos, aplicaciones móviles, sistemas web o simulaciones) combinando o modificando otros existentes.

La creación y/o modificación de artefactos computacionales involucra:

- analizar diferentes ideas, evaluando las ventajas y desventajas de cada alternativa y planificar considerando tiempos y funcionalidades;
- crear artefactos computacionales utilizando entornos para la enseñanza;

- modificar un artefacto computacional existente (por ejemplo, agregar a un videojuego o animación alguna condición particular de finalización), y
- poner a prueba y refinar artefactos computacionales, mediante un proceso deliberado e iterativo que incluye la depuración de errores (identificación y corrección) y la comparación de los resultados obtenidos con los previstos para efectuar ajustes y contempla los cambios en las necesidades y las expectativas de uso.

C. Desarrollar, utilizar y abrir abstracciones

Las abstracciones, en el sentido utilizado en computación, no refieren a la acción de abstraer, sino a construcciones y modelos abstractos que expresan una solución genérica a un conjunto de problemas similares. En programación, estas construcciones se instancian a través de diferentes artefactos computacionales (por ejemplo, procedimientos, funciones, bibliotecas de funciones, etc.)⁶⁰. Denning (2004) menciona el caso de abstracciones en el contexto de los comandos y las define como “objetos abstractos que realizan acciones”. En infraestructura, esta noción aplica a la noción de un dispositivo computacional construido a partir de componentes, pero cuya funcionalidad es posible verla como abstracta (lo que suele llamarse “caja negra”) a pesar de que los componentes específicos que lo componen o sus características pueden afectar esa funcionalidad.

Las abstracciones se desarrollan para crear soluciones computacionales. El uso de abstracciones elaboradas previamente por nosotros u otras personas nos permite evitar repetir el desarrollo de soluciones de partes de un problema que ya fueron resueltas anteriormente y, de esa forma, agilizar el proceso de resolución completo. El uso de soluciones generalizadas y partes de soluciones diseñadas para una amplia reutilización simplifica el proceso de desarrollo al brindar herramientas de más alto nivel para poder abordar la complejidad de un nuevo problema (K-12 Computer Science Framework Steering Committee, 2016: 78). El modo en que se desarrollan las abstracciones es a través del modelado, la descomposición, la generalización y la clasificación.⁶¹

Abrir una abstracción significa analizar el artefacto computacional que la expresa para comprender cómo está implementado, cuáles son sus partes y cómo contribuyen a expresar la solución correspondiente. Esto permite no solamente poder utilizar la abstracción como “caja negra” para resolver una situación problemática, sino también comprender la lógica de construcción de la abstracción, lo que habilita la modificación de dicha abstracción y el aprendizaje necesario para construir nuevas abstracciones.

⁶⁰ La premisa consiste en que no hace falta volver a desarrollar una solución que permita ordenar una secuencia si ya existen artefactos computacionales que lo realizan.

⁶¹ En el Glosario se definen y ejemplifican cada uno de estos términos, bajo la entrada “Abstracciones”.

Esta práctica computacional involucra:

- analizar y utilizar en las soluciones abstracciones predefinidas sin tener en cuenta las especificidades de la implementación (sin conocer cómo fueron desarrolladas, pero sí para qué sirven);
- reconocer diferentes situaciones donde es posible definir abstracciones;
- comprender las ventajas de utilizar abstracciones desarrolladas por terceros que incluyan diferentes recursos tecnológicos;
- abrir abstracciones reconociendo que los artefactos computacionales que utilizan a diario contienen diferentes capas de abstracción que facilitan el uso de los artefactos y cómo se relacionan o comunican las diferentes capas para dar solución a situaciones problemáticas;
- definir abstracciones a partir de identificar y nombrar características relevantes en un planteo de problema y de reconocer que pueden existir soluciones más generales.

D. Identificar y comunicar problemas de dispositivos y artefactos computacionales

Esta práctica se pone de manifiesto frente a problemas que pueden surgir al utilizar diferentes dispositivos y artefactos computacionales. Implica:

- identificar y describir componentes y funciones de dispositivos y artefactos computacionales utilizando la terminología adecuada para poder comunicar sobre su funcionamiento, necesidades, problemas, etc.;
- incorporar estrategias comunes para intentar diagnosticar y resolver problemas que puedan surgir durante la utilización de dispositivos y artefactos computacionales (por ejemplo, ante dos dispositivos parecidos pero con diferencias significativas, poder encontrar pautas comunes de funcionamiento que permitan solucionar problemas de índole similar);
- comprender cómo se relacionan los diferentes componentes de los dispositivos y artefactos computacionales para poder identificar el componente que podría no estar funcionando de la forma esperada;
- identificar el problema o solución que brinda un artefacto computacional específico para poder, ante posibles fallas, investigar y buscar otros que brinden servicios similares.⁶²

⁶² Por ejemplo, poder reemplazar un navegador específico o un programa de edición de textos por otro, si el que se está utilizando no puede llevar adelante cierta tarea.

Glosario de términos

→ Abstracciones

Se refiere a las construcciones y los modelos abstractos que expresan una solución genérica a un conjunto de problemas similares. En programación, se instancian a través de diferentes [artefactos computacionales](#) (por ejemplo, procedimientos, funciones, bibliotecas de funciones, etc.). Denning (2004) define las abstracciones en el contexto de los comandos como: “objetos abstractos que realizan acciones”.

Las abstracciones se desarrollan para crear soluciones computacionales. El uso de soluciones generalizadas y partes de soluciones diseñadas para su reutilización simplifica el proceso de desarrollo al brindar herramientas de más alto nivel para poder abordar la complejidad de un nuevo problema (K-12 Computer Science Framework Steering Committee, 2016: 78).

El modo en que se desarrollan las abstracciones es a través del modelado, la descomposición, la generalización y la clasificación.

El **modelado** consiste en la representación de un elemento, problema o situación del mundo real capturando los aspectos relevantes para un propósito particular y omitiendo cualquier otro detalle. Por ejemplo, una persona puede ser representada a través de su nombre, un número de documento de identidad y su fecha de nacimiento si el objetivo es construir un sistema de identificación unívoca a partir de datos registrados. Cualquier otro detalle de la persona (como el color de ojos, la estatura, etc.) no resultaría relevante para ese propósito. Otro ejemplo es la utilización de figuras geométricas para representar elementos de la realidad: modelar una casa a través del dibujo de la misma.

La **descomposición** consiste en dividir un problema complejo en problemas de menor complejidad, o sea, en subproblemas que sean fáciles de abordar y cuyas soluciones se puedan combinar para formar una solución general. A su vez, cada uno de los subproblemas puede ser descompuesto de la misma forma. Asimismo, esta capacidad también se puede entender en sentido contrario: para analizar un sistema, se puede analizar cada una de sus partes por separado, considerando las relaciones entre ellas y, finalmente, utilizar esa comprensión integradamente para el entendimiento del sistema en su totalidad. Ambos acercamientos son válidos para poder manejar la complejidad. Por ejemplo, la representación de una casa puede descomponerse en diversas partes: dibujar las paredes, el techo, las ventanas, la puerta, etc. A su vez, se puede descomponer el dibujo de la puerta: dibujar un rectángulo con un pequeño círculo para representar el picaporte.

La **generalización** es el reconocimiento de patrones comunes y la capacidad de dar nombres a los esquemas que expresan cada uno de esos patrones. Consiste en encontrar qué características son compartidas por una serie de elementos permitiendo explicitar cuáles los diferencian. Por ejemplo, si se desea construir un procedimiento que dibuje cuadrados de distintos tamaños, este deberá contemplar qué tienen en común los elementos. Así se podrá considerar si la forma de dibujarlos puede ser siempre la misma variando la extensión de los lados de las figuras. En este caso, “ser cuadrado” es la característica que los define y el tamaño es la que los diferencia. Esto permite, por ejemplo, en programación, definir una única tarea que dibuje cuadrados en general y que reciba como dato la extensión de los lados que el usuario quiere darle a cada cuadrado en particular. Así, el mismo código puede utilizarse en otras partes de un programa, pero estar escrito una única vez. Así también cualquier modificación realizada sobre dicho código se verá reflejada en todos los sectores en donde es utilizado. Si no se generaliza la tarea, habría que definir una tarea distinta para dibujar cada cuadrado de distinto tamaño que se desee dibujar y frente a la necesidad de un cambio introducirlo en múltiples lugares del código.

La **clasificación** es entendida como la capacidad de agrupar un conjunto de elementos en diferentes categorías, o sea, definir clases o tipos. Es un caso especial de la generalización que se combina junto con el modelado porque se enfoca en las características comunes de una serie de elementos y, sobre esta base, construye una categoría o un tipo que representa exactamente estas características. Por ejemplo, si consideramos los elementos que constituyen el dibujo de una casa, estos pueden clasificarse según su forma geométrica: triángulos, rectángulos, trapecios, círculos, etc. A su vez, estas categorías pueden agruparse en el tipo de las figuras geométricas, que se distinguen de otras figuras, por ejemplo, de dibujo libre. Otra forma de razonar el proceso de clasificación es reconociendo qué operaciones pueden realizarse sobre una serie de elementos. Por ejemplo, los números pueden ser sumados, restados, multiplicados, divididos, entre otras operaciones, mientras que los colores pueden ser combinados de formas particulares para obtener nuevos colores. Por esta razón, se puede idear un tipo de elemento llamado “número” que represente la posibilidad de realización de las operaciones aritméticas y otro tipo de elemento llamado “color” que represente la posibilidad de realizar operaciones sobre colores.

→ Algoritmo

En esta propuesta consideramos que un algoritmo es una descripción general de un proceso que permite construir una solución para todas las instancias de un problema, por ejemplo, cómo transformar una lista cualquiera de números para que quede ordenada de menor a mayor. Esta concepción es mucho más sofisticada que una secuencia de pasos, por eso en esta propuesta, no consideramos como algoritmos los pasos para resolver tareas cotidianas (como atarse los cordones). Esta decisión es producto de discusiones

del equipo en relación al término y los problemas que su uso simplificado o banalizado incluso en producciones de la propia Iniciativa Program.AR previas.

→ Artefacto computacional

Consiste en una construcción lógica –a veces en combinación con elementos físicos de distintos dispositivos– para solucionar problemas de naturaleza computacional. Algunos ejemplos de artefactos computacionales son los sistemas de *software* y programas específicos, los kits de robótica educativa, los protocolos de comunicación, las metodologías y herramientas vinculadas a soluciones computacionales, etc.

Se distinguen los artefactos de los [dispositivos computacionales](#), que se refieren al aparato físico destinado a llevar adelante cómputos: computadoras de escritorio, teléfonos inteligentes, televisores inteligentes, tabletas, consolas de videojuegos, sistemas de control de automóviles, artefactos hogareños o aparatos industriales, etc.

→ Ciencias de la Computación

Es el nombre del área del conocimiento que estudia las bases y los fundamentos sobre los que se articula la creación y utilización de tecnologías computacionales. Engloba un conjunto de saberes, tales como la programación, el funcionamiento de las redes de computadoras, la representación de la información, entre otros. Resulta fundamental para comprender un mundo cada vez más organizado y dependiente de la tecnología y para poder participar activa, libre y soberanamente en debates actuales sobre la interacción entre las computadoras y la sociedad.

→ Ciudadanía digital

Usualmente, el término denomina el comportamiento, las prácticas y las interacciones entre personas en entornos virtuales (por ejemplo, en salas de chats, juegos en línea, redes sociales, etc.). Frecuentemente, las propuestas de enseñanza referidas al término de “ciudadanía digital” abordan saberes relacionados a la seguridad y la ética informática brindando recomendaciones que dependen de elecciones y responsabilidades individuales. No brindan, por lo general, nociones claves respecto de cómo funcionan los sistemas y las estructuras macrosociales que permiten ataques y vulnerabilidades.

Nosotros proponemos un enfoque que, por un lado, tenga en cuenta que no hay escisión posible entre la ciudadanía digital y el ejercicio de nuestros derechos en otras esferas de la vida: lo que se ve afectado y transformado por la [tecnología digital y computacional](#) es el conjunto de derechos y obligaciones que tenemos como personas y también la cantidad y el tipo de delitos de los que podemos ser víctimas. Por otro

lado, consideramos que, para poder tomar decisiones más o menos informadas y autónomas sobre el uso y modos de uso de dicha tecnología, es necesario conocer su funcionamiento.

→ [Derechos de la ciudadanía asociados a los artefactos y dispositivos computacionales](#)

Los derechos de la ciudadanía asociados a los artefactos y los dispositivos computacionales incluyen cuatro libertades básicas: la libertad de usar el artefacto o el dispositivo con cualquier propósito (**uso**), la libertad de estudiar cómo funciona el artefacto o el dispositivo y ser capaces de modificarlo adaptándolo a las necesidades propias (**estudio**), la libertad de distribuir copias del artefacto o dispositivo para ayudar a otros usuarios (**distribución**) y la libertad de mejorar el artefacto o el dispositivo y hacer públicas esas mejoras para beneficiar a la comunidad (**mejora**). Además, incluye la capacidad de hacer trabajo colaborativo para la construcción de nuevos artefactos y dispositivos, y se vincula con otras cuestiones tales como la propiedad intelectual, la vigilancia en las redes, la neutralidad en la red, circulación de información falsa, etc. Los primeros cuatro derechos se engloban bajo la denominación *software libre* (cuando se habla de artefactos) y *hardware libre* (cuando se habla de dispositivos), y se vinculan con las formas de licenciamiento de estos productos. Sin embargo, al enfocar el tema desde el área de ciudadanía digital, creemos que es mucho más rico hablar de derechos asociados y hacer foco en ellos y en la capacidad de producción colaborativa. Se suele tomar al *software* y al *hardware* libres como un “tema” de estudio que comienza y termina. Nuestra propuesta, en cambio, es usar exclusivamente *software* libre en todas las instancias de la enseñanza para lograr la soberanía tanto pedagógica como tecnológica. Además, se forma al estudiantado con una visión colaborativa, solidaria, federal y nacional, un modelo antagónico al del individualismo, la mercantilidad, la ilegalidad y la dependencia que trae el uso de *software* privativo.

→ [Dispositivo computacional](#)

Denomina al aparato físico destinado a llevar adelante cómputos de diversa naturaleza (por ejemplo, computadoras de escritorio, teléfonos y televisores “inteligentes”, tabletas, consolas de videojuegos, robots, sistemas de control como los que pueden tener los automóviles, artefactos hogareños o aparatos industriales, etc.). Lo diferenciamos de los artefactos computacionales para distinguir la construcción lógica —a veces en combinación con elementos físicos— que se crea para solucionar problemas de naturaleza computacional, por ejemplo, sistemas de *software* y programas específicos, kits de robótica, protocolos de comunicación, metodologías y herramientas vinculadas a soluciones computacionales, etc. (ver [Artefacto computacional](#)).

→ [Habilidades](#)

Se utiliza como la aptitud o la capacidad de hacer una cosa, una actividad que se ejerce, un “saber hacer” general (Roegiers, 2000). Como señala Meirieu (1990): “ninguna capacidad existe en estado puro y toda capacidad se manifiesta a través de la aplicación de contenidos”. Por este motivo, no definimos el desarrollo de una habilidad por sí misma ni la aislamos como tal, ya que entendemos que se ejercitan o ponen en juego en la trama de contenidos específicos que le otorgan contexto (Maggio, 2018). Ejemplos de estas son: modelar, descomponer, generalizar, clasificar. (Se incluye una descripción de lo que implican estas habilidades en el contexto computacional a propósito de la definición de [Abstracciones](#)).

→ [Hardware libre](#)

Ver [Derechos de la ciudadanía asociados a los artefactos y dispositivos computacionales](#).

→ [Huella digital](#)

Cuando uno navega en internet, se recolecta información sobre nuestra interacción (clics, sitios visitados, búsquedas, tiempo de permanencia, ubicación geográfica, etc.) en forma continua, eso constituye un rastro económicamente valioso que se conoce con el nombre de huella digital. La huella digital incluye datos que dejamos involuntariamente, además de los voluntarios.

→ [Inteligencia artificial](#)

En forma cotidiana y/o comercial se suele utilizar el término *Inteligencia Artificial* para denominar dos aspectos completamente diferentes: por un lado, el *aprendizaje automático y la ciencia de datos*, que es el aspecto más prolífico actualmente; y, por el otro, el estudio de la generación de *inteligencia y conciencia en forma artificial*, que actualmente está lejos de poderse concretar y constituye más una meta que una realidad. El término *inteligencia artificial*, en el imaginario popular, tiende a considerar a las máquinas actuales como entidades inteligentes o incluso pensantes, y esto no es (al día de hoy) correcto. Desde el ámbito educativo debemos elegir cuidadosamente la terminología para que cada fenómeno sea correctamente entendido y no utilizar el término más popular por “seguir la moda”. En esta propuesta, usamos los términos de *aprendizaje automático y ciencia de datos* cuando nos referimos a la noción popularizada de inteligencia artificial. Esta decisión se basa en que no queremos contribuir a fomentar la equivocada idea de que las máquinas son entidades inteligentes.

→ [Modelos de cómputo y modelos de arquitectura de computadoras](#)

Un *modelo de cómputo* es una definición de un mecanismo abstracto que modela la ejecución automática de procesos de computación (Aho, 2011). Ejemplos de modelos de cómputo son las máquinas de Turing, las

funciones recursivas, el cálculo lambda, los autómatas celulares, los sistemas de producción, las máquinas de acceso aleatorio (RAM, *random access machines*), entre muchos otros. Por otra parte, un *modelo de arquitectura* de computadoras es un conjunto de técnicas que permiten describir, implementar y construir máquinas programables (combinando diferentes componentes) que puedan ejecutar transformaciones de información en forma efectiva. El modelo de arquitectura más exitoso y difundido es el modelo von Neumann inspirado en el modelo de computación de las máquinas de Turing. La preeminencia del modelo de arquitectura von Neumann tuvo como consecuencia un foco centralizado en las formas de programación basadas en comandos (programación imperativa), relegando otras formas de expresar programas, como las expresiones. Al pensar la computación como una ciencia, debemos tener presente que el modelo de cómputo de las máquinas de Turing o de las máquinas de acceso aleatorio (que modelan la programación basada en comandos) es solamente una de las formas de expresar el procesamiento de la información (si bien una extremadamente importante), y debe ser trabajado como un ejemplo del concepto general de modelos de cómputo.

→ Pensamiento computacional

Este término se utiliza de modos muy diversos y con muy diferentes acepciones en diferentes contextos. Para esta propuesta consideramos la definición ofrecida por Alfred Aho (2011): “los procesos de pensamiento involucrados en la formulación de problemas de forma tal que sus soluciones puedan ser representadas como pasos computacionales y algoritmos”. La idea de algoritmo, como la entiende Aho, está íntimamente ligada a la noción de [modelo de cómputo](#), es decir que estos procesos de pensamiento involucran también la comprensión de los modelos de cómputo y en ocasiones la construcción de nuevos modelos de cómputo, cuando los existentes no son suficientes. Esta forma de entenderlo, tal como discuten Bonello y Schapachnik (2020), liga indisolublemente la noción de pensamiento computacional a la noción de computación y a las Ciencias de la Computación, tal como las presentamos en este documento. Es por todo esto que preferimos no utilizar el término pensamiento computacional y, en su lugar, usamos el término Ciencias de la Computación, que es más rico.

→ Prácticas computacionales

Se refiere al “saber hacer” de computación que se prioriza en la enseñanza obligatoria. Es en el marco de estas prácticas que los conceptos, incorporados en esta propuesta, se construyen, adquieren sentido y se organizan. Las prácticas computacionales, aun cuando se encuentran implicadas en la construcción de los saberes priorizados, se explicitan en forma separada para que el cuerpo docente pueda planificar situaciones de enseñanza a partir de ellas. Las cuatro prácticas que se destacan son:

- reconocer y desarrollar soluciones para problemas computacionales,
- crear y/o modificar [artefactos computacionales](#),
- desarrollar, utilizar y abrir [abstracciones](#),
- identificar y comunicar problemas de [dispositivos](#) y artefactos computacionales.

→ Representación simbólica de datos

En computación, los datos deben representarse en forma simbólica mediante una descripción codificada que pueda distinguirse de otras e identificar de forma unívoca al dato representado. Una forma típica de representación de datos es la representación binaria, que permite representar números utilizando cadenas de ceros y unos. Otra forma de representación es a través de estructuras de datos (tales como registros, listas, etc.) en lenguajes de programación, para modelar entidades y sus atributos. Estas estructuras adoptan diversas formas según los diferentes paradigmas, lenguajes de programación o aplicaciones que los utilizan. Al considerar las CC en sentido amplio, debemos tener en cuenta que la representación binaria es solamente un ejemplo de representación de información y debe ser trabajada como un ejemplo del concepto más general de representación de información. No debe confundirse el concepto de representación de datos con los usos que se les puede dar en otras disciplinas, como las que provienen del área de las Ciencias Sociales, por ejemplo.

→ Representación de la información

Ver [Representación simbólica de datos](#).

→ Robótica

La robótica hace referencia a por lo menos tres campos de estudio:

- A. La **robótica industrial**, que se dedica al diseño, confección y operación de maquinaria que realiza tareas repetitivas en contextos fabriles o similares. Se trata en general de dispositivos electromecánicos controlados digitalmente mediante un ciclo de control relativamente sencillo, más allá de que realicen tareas complejas.
- B. La **robótica general**, que se dedica al estudio, diseño y confección de entidades más bien autónomas, capaces de operar en ambientes no controlados y realizar planificaciones complejas para lograr su objetivo, es decir, un brazo mecánico que suelda partes en una fábrica tiene muy poco que ver con un robot de rescate que se mete entre escombros para buscar y auxiliar a víctimas.

- C. La **robótica educativa**, área donde se utilizan robots para reforzar aprendizajes de diversas disciplinas, por ejemplo, para calcular ángulos de giro en Matemática, temas de Física en el manejo de sensores, etc.

La robótica general cobra relevancia porque está presente en cada vez más áreas de la vida cotidiana. Esa comprensión requiere entender dos conceptos distintos: por un lado, cómo funcionan los sensores y actuadores que interactúan con el mundo físico y, por otro lado, cómo funcionan las computadoras que los controlan. Si consideramos que entre los objetivos de aprendizaje de la materia Física ya se encuentran presentes los conceptos de mecánica y del instrumental necesarios para entender el funcionamiento de un motor o la detección de un obstáculo por el rebote de una onda electromagnética, lo que resta incorporar es el entendimiento de los programas que controlan esos dispositivos (cubierto en los temas de programación). Es decir, teniendo conocimientos de Computación es posible comprender la forma en que funcionan los robots aun sin una formación específica en robótica porque se trata de un caso particular de una disciplina más general. En los años 90, se instaló la idea de que usar *software* de ofimática era el saber necesario que la escuela debía transmitir para que la ciudadanía pudiera participar del mundo digital. Esto se vuelve a expresar en la actualidad a través de la idea de que es imprescindible enseñar robótica. Como herramienta, el uso de robots para el aprendizaje de otros temas (es decir, la robótica educativa) ha mostrado ser efectivo,⁶³ y lo mismo puede decirse de su utilización con niños y niñas para iniciar el aprendizaje de la programación,⁶⁴ ya que son elementos altamente motivadores. Es decir, valoramos su inclusión como un medio más que como un fin.

→ [Software libre](#)

Ver [Derechos de la ciudadanía asociados a los artefactos y dispositivos computacionales](#).

→ [Tecnología digital o computacional](#)

En este trabajo distinguimos el término computacional, que refiere al procesamiento de la información, del de digital, que refiere a una forma de representar información. El término digital hace referencia a una forma específica de representar la información basándose en cadenas de símbolos tomados de una cantidad finita de valores posibles (típicamente se utilizan dos, por lo que se habla de representación binaria) en lugar de las formas analógicas que utilizan un modelo continuo. Por ejemplo, los antiguos discos de vinilo o los casetes son formas analógicas de representar el sonido, mientras que las actuales formas utilizan representaciones

⁶³ Ver, por ejemplo, Fabiane Barreto y Vavassori Benitti (2012), "[Exploring the educational potential of robotics in schools: A systematic review](#)", *Computers & Education*, Vol, 58 (3), abril: 978-988.

⁶⁴ Por ejemplo, Martínez, Gómez y Benotti (2015).

basadas en la representación binaria. Así, al hablar de tecnología digital se está haciendo referencia a un estado particular de la tecnología de los dispositivos y las técnicas asociadas, a su representación y manipulación. El término *tecnología digital* se difundió de manera general y se utiliza muchas veces como sinónimo de *tecnología computacional*. Sin embargo, al pensar en la tecnología computacional hacemos referencia a los procesos lógicos y las formas de expresarlos de las cuales las formas digitales son solamente un ejemplo particular. En este trabajo elegimos el término tecnologías computacionales para poner el foco en los conceptos relevantes.

Referencias bibliográficas

- Aho, A. (2011). "[Computation and Computational Thinking](#)". *Ubiquity*, volumen 2011, enero.
- Bell, T., Tymann, P., y Yehudai, A. (2018). "The big ideas in computer science for K-12 curricula". *Boletín de EATCS*, 1(124).
- Bonello, M.B. y Schapachnik, F. (2020). "Diez preguntas frecuentes (y urgentes) sobre pensamiento computacional". *Virtualidad, Educación y Ciencia*, 11(20): 1-11.
- Centro de Estudios Legales y Sociales [CELS] (2022). "[El ministerio de seguridad de la ciudad buscó información biométrica de 7 millones de personas de manera ilegal](#)".
- Clavero, J. A. (2018). "Posverdad y exposición selectiva a *fake news*. Algunos ejemplos concretos de Argentina". *Contratexto*, 29: 167-180.
- Connell, R. (1997). "La justicia curricular". *Laboratorio de Políticas Públicas*, 6 (27), pp. 1-10.
- Denning, Peter J. (2004). "Great principles in computing curricula". *Boletín de ACM SIGCSE*, 36(1), pp. 336-341.
- Dussel, I. (2007). "La transmisión cultural asediada: los avatares de la cultura común en la escuela". *Propuesta Educativa*, 28: 19-27.
- Guerrero, A. B. (2001). "Estructuración de los contenidos del currículo. Funcionalidad y calidad". *Cuestiones pedagógicas. Revista de Ciencias de la Educación*, 15.
- Goode, J., y Margolis, J. (2011). "Exploring computer science: A case study of school reform". *ACM Transactions on Computing Education [TOCE]*, 11(2): 1-16.
- Maggio, M. (2018). "Habilidades del siglo XX: cuando el futuro es hoy". *XIII Foro Latinoamericano de Educación (Santillana)*.
- Martínez, C.; Gómez, M. J. y Benotti, L. (2015). "A Comparison of Preschool and Elementary School Children Learning Computer Science Concepts through a Multilanguage Robot Programming Platform". *ACM Conference on Innovation and Technology in Computer Science Education [ITiCSE '15]*: 159-164.
- Martínez, C., E, P., Gómez, M. J., Borchardt, M., & Garzón, M. (2022). "[Hacia un currículum emancipador de las Ciencias de la Computación](#)". *Revista Latinoamericana de Economía Y Sociedad Digital*, 3.
- Roegiers, X. (2000). "Saberes, capacidades y competencias en la escuela: una búsqueda de sentido". *Innovación educativa*, 10: 105-122.
- Simari, G. R. (2013). "Los fundamentos computacionales como parte de las ciencias básicas en las terminales de la disciplina Informática". VIII Congreso de Tecnología en Educación y Educación en Tecnología (TE&ET 2013), Santiago del Estero: Red de Universidades con Carreras en Informática (RedUNCI)

Schapachnik, F. y Bonello, M. B. (2022). "Ciencias de la computación en la escuela. Guía para enseñar mucho más que a programar". Colección "Educación que ladra". Buenos Aires: Siglo XXI editores.

Terigi, F. (1999). "Conceptos para el análisis de políticas curriculares. Currículo: itinerario para aprehender un territorio". Buenos Aires: Santillana: 115–136.

Tula, M. I. (2017). "Voto electrónico, controles, resguardos y garantía". En Tullio, A., Pomares, J., y Leiras, M. (2017). *Reforma política en Argentina: un debate democrático pendiente*. Rosario: UNR Editora.

Unesco (2019), [*Descifrar el código: la educación de las niñas y las mujeres en ciencias, tecnología, ingeniería y matemáticas \(STEM\)*](#).

Unesco (2019b), [*Id blush if I could: closing gender divides in digital skills through education*](#).

Unesco (2019c). *Coding, Programming and the Changing Curriculum for Computing in Schools. Report of UNESCO/IFIP TC3 Meeting at OCCE*, 27 de junio.

Zapata-Ros, M. (2018). "Pensamiento computacional. Una tercera competencia clave. El pensamiento computacional como una nueva alfabetización en las culturas digitales". *Revista de Educación a Distancia*, 46 (4).

Documentos analizados para la confección de la propuesta curricular

ACM Committee for Computing Education in Community Colleges (2017). *Computer Science Curricular Guidance for Associate-Degree Transfer Programs with Infused Cybersecurity*. Nueva York: Association for Computing Machinery.

Argentina, Consejo Federal de Educación. Resolución 135/11. [Núcleos de aprendizaje prioritarios de Educación Tecnológica](#), Anexo 3. [En línea].

Argentina, Consejo Federal de Educación. Resolución 343/2018. [Núcleos de Aprendizaje Prioritarios para Educación Digital, Programación y Robótica](#). [En línea].

Association for Computing Machinery, Code.org, Computer Science Teachers Association, Cyber Innovation Center y National Math and Science Initiative (2016). [K-12 Computer Science Framework](#). [En línea].

BCS, Microsoft, Google y Intellect (2012). *Computer Science: A curriculum for Schools*. Reino Unido: Computing at School Working Group.

CSTA (2017). [CSTA Standards](#). [En línea].

Goode, J., y Chapman, G. (2015). *Exploring Computer Science*, versión 6.2.

Hazzan, O., Lapidot, T. y Ragonis, N. (2014). *Guide to Teaching Computer Science*. Segunda edición. Londres: Springer-Verlag.

Ko, A. J. y otros (2020). "It is time for more critical CS education". *Comm.ACM*, 63 (11): 31–33.

K-12 Computer Science Framework Steering Committee (2016). *K12 Computer Science Framework*. Technical Report. Association for Computing Machinery. New York, NY, USA.

Society for Technology in Education [ISTE] (2016). [Standards for Students](#). [En línea].