



**INFORME
DEL GRUPO DE TRABAJO SCIE/CODDII
SOBRE LA ENSEÑANZA PREUNIVERSITARIA
DE LA INFORMÁTICA**

—
JUNIO 2018

El presente informe es el resultado de un trabajo colectivo desarrollado en el seno de la SCIE y la CODDII, al que han contribuido:

Velázquez Iturbide, J. Ángel (coordinador) – Universidad Rey Juan Carlos

Bahamonde Rionda, Antonio – Universidad de Oviedo

Dabic, Sanja – Asociación de Profesores de Informática de la Comunidad Valenciana (APICV)

Escalona, María José – Universidad de Sevilla

Feito, Francisco – Universidad de Jaén

Fernández Cabaleiro, Santiago – Asociación Andaluza de Profesores de Informática (AAPRI)

Ferrero Martín, Begoña – Universidad del País Vasco/Euskal Herriko Unibertsitatea

Garay Vitoria, Néstor – Universidad del País Vasco/Euskal Herriko Unibertsitatea

García, José Carlos – Asociación de Profesores de Informática de la Comunidad Valenciana (APICV)

García Borgoñón, Laura – Universidad de Sevilla

García Martínez, Manuel – Asociación de Profesores de Informática de la Comunidad Valenciana (APICV)

García Molina, Jesús – Universidad de Murcia

García Varea, Ismael – Universidad de Castilla - La Mancha

Hermenegildo Salinas, Manuel – Universidad Politécnica de Madrid

Larraza Mendiluze, Edurne – Universidad del País Vasco/Euskal Herriko Unibertsitatea

Llorens, Faraón – Universidad de Alicante

Mateos, José Andrés – Asociación Andaluza de Profesores de Informática (AAPRI)

Moratel Muñoz, Antonio – Asociación de Profesores de Informática de Castilla-La Mancha (ARPICM)

Mozos, Daniel – Universidad Complutense de Madrid

Pimentel, Ernesto – Universidad de Málaga

Sahelices, Benjamín – Universidad de Valladolid

Toro, Miguel – Universidad de Sevilla

Zapata Ros, Miguel – Universidad de Murcia

Índice

RESUMEN	1
1. Introducción	3
2. Razones para estudiar informática	5
3. Preliminares	7
3.1. Informática y competencia digital.....	7
3.2. Competencia digital	8
3.3. Informática y programación.....	9
3.4. Pensamiento computacional.....	11
3.5. Una nota sobre el aprendizaje de la programación	12
4. Situación en España	14
4.1. Legislación nacional.....	14
4.2. Otras iniciativas	16
5. Propuesta de organización de la educación en informática	17
5.1. Planteamiento global	17
5.2. Áreas y objetivos	18
6. Implantación	25
6.1. Didáctica.....	25
6.2. Formación del profesorado.....	26
Anexo A. Correspondencia de los objetivos de competencia digital con el marco DIGCOMP ...	27
Anexo B. Ejercicios de programación.....	31
Referencias.....	32

RESUMEN

La Sociedad Científica Informática de España (SCIE) y la Conferencia de Directores y Decanos de Ingeniería Informática (CODDII), conscientes de la importancia creciente para las nuevas generaciones de una formación universal en conocimientos básicos de informática, manifiestan la necesidad de incluir en el sistema educativo español la materia “Informática”, de carácter obligatorio desde Educación Primaria hasta Bachillerato.

Competencia en informática

- La competencia digital es una **competencia clave** propuesta por la Unión Europea y reconocida por España. Tener destrezas básicas de informática es una necesidad en la sociedad actual, que se suma a otros conocimientos básicos, como leer, escribir o realizar operaciones aritméticas.
- Una **formación completa** para la sociedad de la información no solamente exige adquirir **competencia digital** sino también los fundamentos conceptuales y metodológicos de la **informática**. Por tanto, una formación completa debe incluir tanto la “**competencia digital**” (conocimientos de informática como usuarios) como la “**ciencia informática**” (el estudio de los principios básicos científicos y técnicos de la informática).
- La enseñanza reglada de la informática tiene un papel cada vez más destacado en los **países de nuestro entorno**, sirviendo de ejemplo los modelos desarrollados e implantados en Inglaterra e Israel. España se encuentra muy retrasada en esta materia y es urgente actuar.

Materia “Informática”

- El desarrollo de cualquier competencia clave no transversal exige cursar **asignaturas independientes y de carácter obligatorio**, papel que cumplen las asignaturas de Lengua, Matemáticas, Física, Química o Ciencias Naturales para sus correspondientes competencias.
- Los alumnos deben cursar una asignatura obligatoria por curso de la **nueva materia “Informática”**, desde Educación Primaria hasta Bachillerato pasando por Educación Secundaria Obligatoria. Esta materia debe reglarse legislativamente de forma que se detallen sus objetivos, contenidos, criterios de evaluación y estándares de aprendizaje evaluables, como en el resto de asignaturas.
- La informática constituye una disciplina mixta con elementos de ciencia y tecnología. Por tanto, su aprendizaje exige una **asignatura independiente** de otras con las que tiene relación pero cuyos contenidos son distintos, como Matemáticas o Tecnología.
- Los **contenidos** de competencia digital y de ciencia informática deben variar según las etapas educativas y el correspondiente grado de madurez de los alumnos:
 - **Educación Primaria.** Debe incluir contenidos de competencia digital y conocimientos básicos de programación, computadores, redes y datos. La

competencia digital debe asegurar unos conocimientos y habilidades básicos para el uso eficaz, seguro y ético de las tecnologías de la información.

- **Educación Secundaria Obligatoria.** Es similar, pero los conocimientos, sobre todo de programación, deben adquirir un carácter más abstracto. La competencia digital se desarrollará a un nivel de usuario medio.
 - **Bachillerato.** Debe distinguirse entre las modalidades de Ciencias, Ciencias Sociales y Humanidades, y Artes. En todos los casos se profundizará en conocimientos de ciencia informática y se resaltará la aplicación de herramientas informáticas en su ámbito.
- Los **modelos implantados en Inglaterra y en Israel** son excelentes fuentes de inspiración para los distintos aspectos de su implantación y podrían adaptarse a nuestro sistema educativo.

Implantación de la materia “Informática”

- La impartición de la materia “Informática” en asignaturas obligatorias **facilitaría su uso transversal en otras asignaturas**, al igual que sucede con Lengua, Matemáticas o Inglés. Un buen ejemplo es la educación bilingüe en inglés, cuyo uso transversal en una mayoría de asignaturas se compatibiliza con una asignatura específica de Inglés.
- Un aspecto clave para su implantación exitosa es la **formación del profesorado**. Las administraciones deben asegurar una formación adecuada en:
 - Competencia digital. El profesorado debe avanzar hacia un nivel adecuado de competencia digital. Juegan un papel clave los cursos de formación para el profesorado en plantilla y los Grados y Posgrados de Educación para el futuro profesorado.
 - Informática. La impartición de esta materia exige un perfil propio e independiente del actual perfil de Tecnología, que se centra en otras tecnologías muy alejadas de la informática y cuya formación es distinta.

Beneficios personales, sociales y nacionales

- La formación recibida permitirá que todos los alumnos tengan una educación básica en informática que les capacite para su mejor desarrollo profesional futuro. Se deja a la **Universidad** tanto el desarrollo de la competencia a nivel de experto dentro de las necesidades de cada Grado como el desarrollo del conocimiento científico e ingenieril de los alumnos interesados en Ingeniería Informática y Grados afines.
- La formación en informática desde edades tempranas permite la familiarización con un campo en el que las **mujeres** tienen un papel minoritario. El acercamiento a la informática desde la infancia puede fomentar un mayor porcentaje de vocaciones femeninas.
- Una población formada en informática favorecerá un mayor **desarrollo empresarial** de los sectores económicos con valor añadido y, en suma, un **cambio de modelo económico**, así como **mayor competitividad**.
- No debe perderse de vista el actual **entorno internacional competitivo**, donde otros países ya han dado el paso hacia una formación reglada en informática (como Inglaterra o Israel) y el de otros países que están en proceso (tanto dentro de Europa Occidental y Oriental como en los países anglosajones y Extremo Oriente).

1. Introducción

Durante el siglo XXI se ha ido formando una corriente de opinión internacional favorable al aprendizaje de la informática en la educación preuniversitaria. Es el resultado de un proceso lento que se inició a finales del siglo XX. Un momento clave fue la aparición del ordenador personal, a finales de los años 70 y comienzos de los 80. Si antes los ordenadores sólo estaban accesibles a grandes empresas e instituciones, desde ese momento pasan a estar al alcance de ciudadanos, profesionales y pequeñas empresas. Aparecieron nuevas marcas, dispositivos y lenguajes de programación que revolucionaron la informática y nuestra relación con ella. A aquella época se remontan nombres como Spectrum, BASIC, Apple o Windows, de los que algunos han sobrevivido y otros no.

En aquellos años surgió el lenguaje de programación LOGO, orientado a los niños [51]. Con este lenguaje, Seymour Papert presentaba una nueva teoría del aprendizaje, el construccinismo. Según el construccinismo, el conocimiento se forma mejor por medio de la construcción de “artefactos”; en el caso de Logo, por medio de la construcción de programas. Sin embargo, esta iniciativa y otras parecidas no tuvieron el impacto que deseaban sus propulsores.

Desde entonces ha habido numerosos desarrollos técnicos, en hardware, software y comunicaciones, que han universalizado el uso de la informática y han dado lugar a la sociedad de la información. Numerosos sectores productivos tienen sus procesos o sus servicios automatizados. Las administraciones también han fomentado la atención electrónica a la ciudadanía. El uso de nuevos dispositivos digitales se ha extendido a prácticamente todas las edades, de niños a mayores.

En esta situación, el conocimiento de la informática ya era prácticamente un imperativo individual y colectivo hace diez años. Hubo entonces varios sucesos que reactivaron el interés por el aprendizaje de la informática en los colegios e institutos, de los que destacamos dos. Por un lado, la aparición de Scratch, un lenguaje de programación visual basado en bloques y diseñado para motivar a los jóvenes. El lenguaje combinaba una gran facilidad de uso y de aprendizaje con grandes posibilidades para construir programas llamativos, con gráficos, sonidos y animaciones. Según una metáfora de sus propios desarrolladores, Scratch ofrece “un suelo bajo, un techo alto y habitaciones espaciaosas” [53].

Por otro lado, Jeannette Wing, entonces directora adjunta del *Directorate for Computer & Information Science & Engineering (CISE)* de la *National Science Foundation (NSF)*, lanzó un manifiesto a favor de que se divulgara la forma de afrontar y resolver problemas por parte de los informáticos, el “pensamiento computacional” [60]. Wing aseguraba que dicho pensamiento supondría una ventaja para cualquier persona. Además, habría que considerarlo una habilidad tan básica como saber leer, escribir o hacer operaciones aritméticas, por lo que debería enseñarse en el colegio.

Ambos sucesos han catalizado una fuerte tendencia global hacia la inclusión de la informática en los estudios preuniversitarios. Dos decisiones destacadas han sido la iniciativa *Computer Science for All*, del expresidente Obama en EEUU, y el programa *Computing at School*, de Inglaterra en la Unión Europea. La primera ha servido de catalizador y aglutinador en un país con la educación descentralizada [34], destacando la elaboración del marco *K-12 Computer*

Science Framework [2]. La segunda, en Inglaterra, ha producido un gran desarrollo normativo [14][42].

La respuesta de los países europeos es muy variada de un país a otro [11][23], encontrándose desde países sin ninguna educación en informática hasta otros con asignaturas específicas de informática desde Educación Primaria, pasando por países con una implantación transversal o con modelos mixtos. España se encuentra muy atrasada en esta materia y es urgente actuar. La legislación educativa nacional reconoce la competencia digital como una competencia clave [18], se identifica como objetivo en las tres etapas educativas y se proclama su uso transversal [16][17]. Solamente existen algunas asignaturas específicas en Educación Secundaria y en Bachillerato [17], así como asignaturas de libre configuración ofertadas en algunas comunidades autónomas [40].

La Sociedad Científica Informática de España (SCIE) y la Conferencia de Directores y Decanos de Ingeniería Informática (CODDII) propugnan una educación en informática universal y de calidad. En este informe, ambas entidades hacen una recomendación para la educación en informática del alumnado de las etapas educativas de Educación Primaria y Secundaria. La recomendación se basa en los conocimientos adquiridos internacionalmente tras décadas de investigación sobre enseñanza universitaria de la informática, en las recomendaciones de asociaciones internacionales de informática y en las acciones llevadas a cabo en otros países. El objetivo es proporcionar una opinión razonada del mundo académico español en informática que pueda ser útil a políticos, educadores, medios de comunicación y la sociedad en general.

La estructura del informe es la siguiente. En las secciones 2, 3 y 4 presentan algunas cuestiones necesarias para comprender la propuesta presentada: respectivamente, las principales razones para promover la enseñanza preuniversitaria de la informática, varias cuestiones clave sobre la educación en informática, y la situación en España de la enseñanza de la informática. La sección 5 presenta la propuesta educativa de SCIE/CODDII. La sección 6 trata algunas cuestiones prácticas importantes para una implantación exitosa de la materia de Informática. Por último, se incluyen 2 anexos con dos cuestiones importantes pero específicas: la correspondencia de los objetivos propuestos de competencia digital con el marco DIGCOMP y una gama de ejercicios adecuados para el aprendizaje de la programación.

2. Razones para estudiar informática

Las sociedades occidentales actuales son complejas y por tanto los argumentos a favor de la implantación de la educación en informática en el colegio y el instituto son variados. Presentamos las razones más destacadas, sin ser una lista exhaustiva (para otras razones, léase [57]). Básicamente, podrían dividirse en razones económicas y educativas.

No debe sorprender que algunas de las razones más frecuentemente esgrimidas sean de tipo económico. Por un lado, existe un entorno internacional donde el sector cuaternario, relacionado con la información, tiene un peso cada vez mayor. En este contexto, es difícil concebir un desarrollo empresarial, especialmente en los sectores productivos con mayor valor añadido, sin participación de la informática. Se trata de un factor clave para el fomento de la innovación tecnológica y la competitividad. Asimismo, es necesario disponer de mano de obra adecuadamente formada para satisfacer la demanda. Aunque la Formación Profesional y la Universidad proporcionan una formación especializada para el alumnado interesado en los Ciclos Formativos, Grados y Posgrados en Ingeniería Informática o afines, en las etapas educativas anteriores debe proporcionarse una educación básica en informática que sea universal y que capacite para un adecuado desarrollo profesional en el futuro.

Considerando que hoy en día prácticamente cualquier profesión se beneficia de un uso más o menos específico de la informática, pueden distinguirse, al menos, tres perfiles de alumnos en su relación con ella. Una gran parte del alumnado no tendrá mayor relación con la informática que la de ejercer de usuarios. Estos alumnos deben tener una formación básica en el uso de la informática que les permita desenvolverse de forma eficaz, segura y ética. En el otro extremo, hay un alumnado que puede estar interesado en estudiar un Ciclo Formativo, Grado o Posgrado en Ingeniería Informática o afín. Es obvio que a este perfil le interesa tener una formación mínima en informática, que les permita tomar una decisión fundada sobre sus estudios futuros. En medio se sitúa una buena parte del alumnado, sin que exista una división nítida entre informáticos y usuarios de la informática. Un buen número de profesiones necesitan personal formado en usos de la informática pero sin exigir una especialización informática, por ejemplo, la programación de consultas a hojas de cálculo o bases de datos o el uso de herramientas de minería de datos. Por esta razón, numerosos grados ofrecen una asignatura de “informática”. Esta asignatura suele incluirse con el objetivo de ofrecer a los alumnos una competencia digital útil para su futura profesión. Sin embargo, en la práctica los profesores se encuentran con la necesidad de proporcionar una competencia digital básica a muchos alumnos, que carecen de ella. Si éstos la hubieran adquirido en las etapas educativas anteriores, estas asignaturas podrían centrarse en las necesidades de su profesión.

Un potencial beneficio de la inclusión de la educación en informática desde edades tempranas es que un mayor conocimiento de la disciplina podría fomentar las vocaciones de grupos que tradicionalmente tienen una presencia minoritaria. Este problema da con especial gravedad entre las mujeres. Si ya la presencia de mujeres en ingeniería es minoritaria, todavía es más baja en informática. Una mayor familiarización con la informática facilitaría un mejor conocimiento de ésta, despojándola de los estereotipos poco atractivos que hay sobre los informáticos y que no se corresponden con la realidad del ejercicio de la profesión. Esta situación facilitaría que hubiera un mayor porcentaje de vocaciones femeninas en informática. También influiría en la población masculina que, en múltiples ocasiones, llega a los estudios de informática con expectativas equivocadas.

Un segundo bloque de razones es de tipo educativo, sobre todo en relación con el fomento del pensamiento computacional. Si se trata de una habilidad básica que todo ciudadano debería tener en nuestra sociedad y que debería estar integrada en la educación obligatoria. Conviene avisar de que las propuestas no siempre presentan la claridad que sería deseable para una adecuada planificación educativa. Dos problemas que frecuentemente aparecen son la imprecisa definición de pensamiento computacional y la insuficiente distinción entre pensamiento computacional y competencia digital. En nuestra propuesta abordamos ambos problemas para darles una respuesta lo más clara posible.

La informática también ofrece oportunidades para la actualización y mejora de la educación en general. El uso de presentaciones tipo PowerPoint, Internet o las plataformas de e-learning han dejado de estar limitadas a la educación universitaria, estando actualmente integradas en los institutos y avanzando su uso en los colegios. El uso de la informática también ofrece posibilidades para renovar las experiencias educativas y las prácticas en las asignaturas. Esta posibilidad se amplía con los lenguajes de programación, como Scratch o Python, que permiten generar contenidos multimedia y automatizar procesos. Para hacer un uso más provechoso de este uso transversal de la informática, los alumnos deben recibir una formación adecuada. Dicha formación debe rehuir de las modas o las puras novedades tecnológicas y debe basarse en conceptos y prácticas claros.

3. Preliminares

En esta sección se realiza una presentación de los conceptos más utilizados (competencia digital, informática, programación y pensamiento computacional), así como una nota sobre las razones por las que la programación puede ser más o menos difícil de aprender.

3.1. Informática y competencia digital

Existen dos materias relacionadas pero distintas que conviene definir con cierta precisión para evitar confusiones. Ambas han surgido en la exposición anterior. Nos ayudamos de las definiciones informales que da *ACM Europe* [41]:

- Alfabetización digital (*digital literacy*): “Trata del uso del software y los ordenadores. La capacidad de trabajar con ellos e Internet, buscar información, comportarse en la red”. En este informe usaremos el término “competencia digital”, al ser el usado por el Ministerio de Educación, siguiendo las directrices comunitarias.
- Informática (*informatics*): “Es la ciencia que hay detrás de las tecnologías de la información”. Otros términos usados son computación (*computing*) o ciencias de la computación (*computer science, computing science*), pero usaremos informática por su popularidad en España, su precisión y fácil comprensión.

Es fácil apreciar la diferencia entre ambos conceptos. La adquisición de la competencia digital busca que el alumnado sea competente en el uso de la informática, es decir, que sepa manejar los dispositivos digitales (ordenadores, tabletas, consolas de juegos, teléfonos móviles, etc.) y sus programas. Son conocimientos de carácter práctico, que no precisan unos grandes conocimientos conceptuales o procedimentales. La adquisición de esta segunda clase de conocimientos es el objetivo de la informática propiamente dicha, que facilitaría un uso más eficaz de la competencia digital.

Se puede determinar si cierto contenido educativo corresponde a competencia digital o a informática si la respuesta a la siguiente pregunta es respectivamente positiva o negativa:

“¿Puede hacerlo (o aprenderlo) una persona que tenga formación adecuada en el uso de las TIC aunque no haya estudiado informática?”

En general, basta con esta pregunta. Sin embargo, la frontera entre ambas disciplinas no es nítida, por lo que puede ser necesario reformular la pregunta. Por ejemplo, ante el aprendizaje de programar con Scratch, puede preguntarse:

“¿Se enseñan los conceptos o métodos subyacentes a la programación?”

Si se enseñan explícitamente los conceptos de programación (instrucción, secuencia, bucle, paralelismo, etc.), podemos hablar de enseñanza de informática. Si simplemente se enseña a programar en Scratch de manera informal e intuitiva, nos mantendríamos en la competencia digital, con Scratch como una forma sofisticada de producir contenido digital.

Pueden hacerse distinciones con otros términos. Por ejemplo, el programa *Computing at School* de Inglaterra [42] distingue entre *digital literacy*, *information technology* y *computer science*. Dado que las definiciones de los dos primeros términos aluden al uso de los ordenadores, consideramos más clara su fusión. Para el lector interesado, Ala-Mutka presenta detalladamente esta familia de conceptos [8].

3.2. Competencia digital

El principal marco usado en la Unión Europea para definir la competencia digital es el marco DIGCOMP. Se publicó en 2013 [31] y posteriormente se han elaborado las actualizaciones 2.0 [58] y 2.1 [21].

El marco DIGCOMP define la competencia digital mediante 5 “dimensiones”:

1. Áreas competenciales que agrupan a las competencias.
2. Nombre y descriptor de cada competencia.
3. Nivel de maestría para cada competencia.
4. Conocimientos, habilidades y actitudes propias de cada competencia.
5. Ejemplos de uso de cada competencia para distintos propósitos.

Para una mejor comprensión, reproducimos a continuación las dimensiones 1 y 2, respectivamente numeradas AC1-AC5 y C1.1-C5.4:

AC1. Alfabetización en información y datos

- C1.1. Navegar, buscar y filtrar datos, información y contenido digital
- C1.2. Evaluar datos, información y contenido digital
- C1.3. Manejar datos, información y contenido digital

AC2. Comunicación y colaboración

- C2.1. Interactuar mediante tecnologías digitales
- C2.2. Compartir mediante tecnologías digitales
- C2.3. Participar en la ciudadanía mediante tecnologías digitales
- C2.4. Colaborar mediante tecnologías digitales
- C2.5. Comportamiento digital
- C2.6. Gestionar la identidad digital

AC3. Creación de contenido digital

- C3.1. Desarrollar contenido digital
- C3.2. Integrar y reelaborar contenido digital
- C3.3. Derechos de autor y licencias
- C3.4. Programar

AC4. Seguridad

- C4.1. Proteger dispositivos digitales
- C4.2. Proteger datos y la intimidad personales
- C4.3. Proteger la salud y el bienestar
- C4.4. Proteger el medio ambiente

AC5. Resolución de problemas

- C5.1. Resolver problemas técnicos
- C5.2. Identificar necesidades y respuestas tecnológicas
- C5.3. Usar las tecnologías digitales con creatividad
- C5.4. Identificar lagunas de competencia digital

Obsérvese que estas competencias tienen poco contenido conceptual de informática. Para la mejor comprensión de las áreas competenciales y sus competencias asociadas, conviene consultar su definición porque algunos términos se usan en un sentido distinto del usual en

informática. En concreto, la competencia “programar” se presenta en un sentido más amplio, que incluye la configuración de aplicaciones. Asimismo, el área competencial 5 no trata la resolución de problemas en un sentido matemático o algorítmico, sino los problemas prácticos que encuentra el usuario de las tecnologías de la información.

La tercera dimensión del marco DIGCOMP define los niveles de maestría que puede alcanzarse en cada competencia. El marco DIGCOMP 1.0 [31] establecía 3 niveles de maestría que la versión 2.1 [21] ha ampliado a 8, agrupados en 4 pares:

1-2: Nivel básico.

3-4: Nivel intermedio.

5-6: Nivel avanzado.

7-8: Nivel especializado.

La dimensión 4 de DIGCOMP (conocimientos, habilidades y actitudes propias de cada competencia) no se define en DIGCOMP 2.1, desafortunadamente, aunque sería de gran utilidad.

Por último, la dimensión 5 consiste en ejemplos de uso de cada competencia. El marco DIGCOMP 2.1 incluye dos ejemplos para un solo nivel de maestría de cada competencia, un ejemplo del mundo laboral y otro del educativo.

Una consideración importante es que la competencia digital puede desglosarse según el actor a quien se refiera. Por esta razón, puede distinguirse entre la competencia digital alcanzada por los centros educativos, por el profesorado o por el alumnado¹. El objetivo de este informe es centrarnos en la competencia digital del alumnado, reconociendo la importancia de las otras formas de competencia digital. De hecho, la competencia digital del profesorado se tratará brevemente en la Sección 6.2.

El marco DIGCOMP presenta varias ventajas. En primer lugar, se trata de un marco que se ha constituido en un estándar *de facto*. En segundo lugar, presenta un campo tan dispar de forma amplia y bien estructurada, especialmente en sus dimensiones 1 y 2. Desafortunadamente, tiene un carácter genérico similar a los objetivos de las etapas educativas [16][17], proporcionando solamente orientaciones generales para su uso educativo. Por ejemplo, las variaciones presentadas en la dimensión 3 (nivel de maestría) de la versión 2.1 no recogen gradación en la complejidad de objetivos o contenidos, sino que se limitan a diferenciar circunstancias educativas (normalmente, si los problemas están bien definidos, si se dispone de la ayuda del profesor o si la tarea debe realizarla el alumno o enseñársela a otros).

3.3. Informática y programación

Como consecuencia de una larga tradición de reflexión y debate dentro del mundo académico informático, existen diversas definiciones de la disciplina de la informática. Grandes informáticos han contribuido a la creación de una identidad informática, por ejemplo, tratando de identificar las “grandes ideas” que subyacen a la Informática [27][55].

¹ En España, el Instituto Nacional de Tecnologías Educativas y de Formación del Profesorado (INTEF) está abordando las distintas formas de competencia digital.

La informática se ha definido implícitamente mediante los planes de estudios universitarios, entre los cuales los de ACM y la IEEE Computer Society han sido los más influyentes. La unión de fuerzas de ambas asociaciones permitió el desarrollo de un hito en la definición de la informática, el Informe Denning [27]. Este informe situó a la informática en la confluencia de tres tradiciones intelectuales: matemática, científica e ingenieril. También propuso una definición de informática, e identificó nueve subáreas y un cuerpo de conocimientos nucleares. Por último, llamó la atención sobre la importancia de los laboratorios. Reproducimos su definición de informática [25, pág. 12]:

“La disciplina de la informática (*computing*) es el estudio sistemático de procesos algorítmicos que describen y transforman información: su teoría, análisis, diseño, eficiencia, implementación y aplicación. La pregunta fundamental que subyace a toda la informática es «¿qué puede automatizarse (eficientemente)?»”

El Informe Denning fue la base para la elaboración de la primera recomendación curricular conjunta de ACM e IEEE Computer Society, el *Computing Curricula 1991* [56]. Estas recomendaciones se han ido actualizando en 2001, 2008 y 2013 [1]. La última versión, del año 2013 [6], nos permite ver la amplitud y precisión de su definición de informática, al reconocer 18 subáreas. Reproducimos las 17 subáreas cuya inclusión en los planes de estudios se considera necesaria total o parcialmente [6, pág. 37]:

- AL – Algoritmos y complejidad
- AR – Arquitectura y organización
- CN – Ciencia computacional
- DS – Estructuras discretas
- GV – Gráficos y visualización
- HCI – Interacción persona-ordenador
- IAS – Aseguramiento y seguridad de la información
- IM – Gestión de la información
- IS – Sistemas inteligentes
- NC – Redes y comunicaciones
- OS – Sistemas operativos
- PD – Computación paralela y distribuida
- PL – Lenguajes de programación
- SDF – Principios de desarrollo de software
- SE – Ingeniería del software
- SF – Principios de sistemas
- SP – Cuestiones sociales y práctica profesional

Todas las recomendaciones curriculares han destacado el papel central de la programación en la informática [27]. Sin embargo, también han aclarado que la informática es una disciplina más amplia, que incluye actividades que no son programación. Por tanto, una formación en informática debe incluir la programación como elemento central pero no único. Este aspecto, claramente reconocido en la educación universitaria, aún no tiene un reconocimiento suficiente en las etapas educativas anteriores, donde suele ponerse el énfasis en la programación (bajo diversos nombres, como código, pensamiento computacional, etc.) o en otras materias llamativas pero tangenciales (como la robótica o la impresión 3D).

3.4. Pensamiento computacional

En la introducción se identificó el pensamiento computacional como una de las propuestas que más han contribuido a difundir la idea de que la informática debe impartirse en las etapas educativas preuniversitarias. En este sentido, se trata de una iniciativa común con la aquí defendida. Tiene una virtud a destacar: se trata de un término nuevo que resulta llamativo y muy motivador para muchos educadores (véanse las recopilaciones de experiencias en [47][49]). Desafortunadamente, es un término poco preciso y que no permite centrar fácilmente el debate educativo. Dado su gran impacto y aunque no es éste el lugar adecuado para un análisis en profundidad, es pertinente hacer algunas aclaraciones.

Una de las definiciones más citadas de pensamiento computacional es la que realizó la propia Jeannette Wing [61]:

“El pensamiento computacional son los procesos de pensamiento que se desarrollan al formular problemas y sus soluciones de forma tal que sus soluciones se representan de una forma que puede ser realizada eficazmente por un agente de procesamiento de información.

Informalmente, el pensamiento computacional describe la actividad mental que formula un problema para que admita una solución computacional.”

La definición es parecida a la dada en el Informe Denning sobre la informática, salvo en dónde pone el énfasis. Mientras que el Informe Denning enfatiza lo que debe estudiarse para aprender informática, el pensamiento computacional enfatiza el “tipo de pensamiento” asociado, algo mucho más difícil de concretar para organizar las enseñanzas.

La imprecisión de la definición del pensamiento computacional es un quebradero de cabeza para gran parte del profesorado [25][26]. Su definición puede equipararse a un objetivo de una etapa educativa, que es un enunciado general, útil para la sociedad pero más limitado para el profesorado, que necesita un desarrollo más preciso (contenidos, criterios de evaluación, etc.) [18].

Diversas personas y organizaciones han realizado propuestas más concretas de pensamiento computacional, como la “operacionalización” ofrecida conjuntamente por la *Computer Science Teachers Association* (CSTA) y la *International Society for Technology in Education* (ISTE) [24]. Por desgracia, esta y otras propuestas destacadas (p.ej. [12][20][22]) no alcanzan el rigor del Informe Denning o los *Computing Curricula*, por lo que han recibido diversas críticas:

- Insuficiente comprensión de la informática y sus conceptos [25][30] (p.ej. algoritmo [7] o codificación [10]).
- Transferencia indemostrada de competencias a otros dominios [26][37].
- Asunción de un orden inverso de alcance de la maestría [26]. Tradicionalmente se ha supuesto que el aprendizaje y práctica de una disciplina produce un aumento de las capacidades y la maestría del alumno. Sin embargo, se argumenta que adquirir “pensamiento computacional” (sin concretarse cómo) facilitará el aprendizaje de la programación.
- Preeminencia del pensamiento computacional sobre otras clases de pensamiento [26][37].
- Presentación del pensamiento computacional de forma ampulosa [25][37], con unas competencias que pueden resultar abrumadoras. Los objetivos propuestos para otras

disciplinas tienen objetivos más modestos, consistentes en el desarrollo por los alumnos de una formación básica y universal basada en competencias específicas.

En resumen, los objetivos de enseñar informática y desarrollar el pensamiento computacional son coincidentes en gran medida. El término “pensamiento computacional” ha tenido el mérito de haber despertado gran interés, pero es preferible utilizar otros marcos más claros para una planificación educativa. Es necesario establecer unos objetivos educativos claros y realistas para el alumnado de Educación Primaria, Secundaria y Bachillerato.

3.5. Una nota sobre el aprendizaje de la programación

Como hemos comentado en las secciones anteriores, la programación ocupa un lugar central en la enseñanza de la informática y el pensamiento computacional. Sin embargo, se da una situación paradójica. Por un lado, se sabe que aprender a programar en la universidad es difícil [44][48]. Uno de los consensos que se vienen forjando últimamente en la comunidad investigadora es que el problema quizá resida en una enseñanza que exige al alumno un esfuerzo cognitivo excesivo [46][59]. Sin embargo, en los últimos años encontramos que los niños aprenden a programar con lenguajes de bloques tipo Scratch con pasmosa facilidad.

La explicación de esta aparente paradoja es que la programación con lenguajes de bloques ha reducido drásticamente el esfuerzo necesario para aprender a programar. Hay varios factores que han contribuido a esta reducción del esfuerzo:

- No es necesario aprender el léxico y sintaxis del lenguaje de programación, sino simplemente reconocer visualmente los bloques del lenguaje.
- Los objetivos de los programas Scratch exigen menos esfuerzo de abstracción que los programas convencionales. Los primeros tienen objetivos de carácter narrativo, visual y acústico, mientras que los segundos suelen resolver problemas de carácter algorítmico.
- La realimentación visual y sonora inmediata facilita la creación y depuración de los programas y aumenta la motivación de los alumnos.
- Los lenguajes tipo Scratch tienen “suelo bajo, techo alto y habitaciones espaciosas” [51][53], es decir, permiten un comienzo sencillo y además ofrecen oportunidades para desarrollar proyectos cada vez más complejos y de diversos tipos.
- En el caso de Scratch, existe una comunidad web que ofrece la oportunidad de colaborar y compartir, lo cual es especialmente motivador para niños y jóvenes.

Sin embargo, estas ventajas no deben ocultar dificultades o riesgos en la enseñanza de la programación en estas etapas educativas:

- Aprendizaje superficial del lenguaje de programación, sin comprensión de los conceptos de programación subyacentes y de los métodos de trabajo aconsejables.
- Ausencia de herramientas de programación (p.ej. un depurador) o disponibilidad de herramientas simples (p.ej. el editor de Scratch), que dificultan el desarrollo de programas no triviales.

- Énfasis en la “creatividad”, frente al rigor y el espíritu crítico. Si se observa la didáctica de disciplinas tradicionalmente creativas (p.ej. las artes), puede verse que se enfatiza la práctica de su faceta técnica, dejando la exhibición de la creatividad a los expertos.
- Ausencia de guía. Con frecuencia se deja que los niños aprendan solos a programar mientras ejercitan “su creatividad”. Se presupone que el lenguaje y el entorno de programación son tan intuitivos que no hace falta enseñarlos explícita y gradualmente. Esta situación puede producir frustración y desmotivación en los alumnos que encuentren dificultades.

En resumen, sería deseable que se desarrollara una didáctica de la programación con el mismo cuidado que en otras disciplinas. Los conceptos y prácticas deben explicarse teniendo en cuenta la edad y el progreso de los alumnos. Obviamente, estas precauciones deben extremarse con lenguajes de programación textuales.

4. Situación en España

Presentamos de forma resumida la situación en España. En primer lugar, se presenta la legislación educativa para después resumir otras iniciativas al margen de la educación formal.

4.1. Legislación nacional

La legislación española recoge las competencias que se consideran clave para todo ciudadano [18] y que, por tanto, deberían promoverse durante los ciclos educativos. Se identifican 7 competencias clave, de las cuales la tercera es la competencia digital (Artículo 2.2):

1. Comunicación lingüística.
2. Competencia matemática y competencias básicas en ciencia y tecnología.
3. Competencia digital.
4. Aprender a aprender.
5. Competencias sociales y cívicas.
6. Sentido de iniciativa y espíritu emprendedor.
7. Conciencia y expresiones culturales.

La descripción de la competencia digital consta de dos partes que coinciden, en lo básico, con la descripción de esta competencia dada por la Unión Europea [50] y con las áreas competenciales del marco DIGCOMP [21].

Obsérvese que la separación de las “competencias básicas en ciencia y tecnología” y la “competencia digital” muestra el carácter instrumental de esta última. Coherentemente con esta relación de competencias, el tratamiento de la informática en las etapas de Educación Primaria [16], Educación Secundaria Obligatoria [17] y Bachillerato [17] se limita a la competencia digital. A continuación, presentamos los objetivos correspondientes y las asignaturas que la desarrollan en cada etapa educativa.

4.1.1. Educación Primaria

En Educación Primaria [16], la competencia digital se menciona explícitamente como objetivo (pág. 19.354):

“i) Iniciarse en la utilización, para el aprendizaje, de las Tecnologías de la Información y la Comunicación desarrollando un espíritu crítico ante los mensajes que reciben y elaboran.”

El decreto no prevé el desarrollo de la competencia digital en asignaturas específicas, sino como un “elemento transversal” (pág. 19.356):

“1. Sin perjuicio de su tratamiento específico en algunas de las asignaturas de cada etapa, la comprensión lectora, la expresión oral y escrita, la comunicación audiovisual, las Tecnologías de la Información y la Comunicación, el emprendimiento y la educación cívica y constitucional se trabajarán en todas las asignaturas.”

4.1.2. Educación Secundaria Obligatoria

En Educación Secundaria Obligatoria, se establece el siguiente objetivo [17] (pág. 177):

“e) Desarrollar destrezas básicas en la utilización de las fuentes de información para, con sentido crítico, adquirir nuevos conocimientos. Adquirir una preparación básica en el campo de las tecnologías, especialmente las de la información y la comunicación.”

En este caso, parece que la competencia digital está ligada a la búsqueda documental.

En el primer ciclo de Educación Secundaria Obligatoria, no existe ninguna asignatura dedicada específicamente a desarrollar la competencia digital. Sin embargo, puede encontrarse la materia específica de opción “Tecnología”, organizada en 5 bloques y donde el bloque 5 es “Tecnologías de la Información y las Comunicaciones” (págs. 529-531).

En el segundo ciclo de Educación Secundaria Obligatoria, existe la asignatura específica “Tecnologías de la Información y la Comunicación” (pág. 532), cuya oferta depende de cada Administración Autonómica y de cada centro educativo. Esta asignatura forma parte de la materia “Tecnologías de la Información y la Comunicación” (págs. 531-534), que se ofrece durante este curso y los dos de Bachillerato.

En la opción de enseñanzas aplicadas, también existe la asignatura troncal no obligatoria “Tecnología” (págs. 451-453). La asignatura consta de 6 bloques, siendo el bloque 1 “Tecnologías de la información y la comunicación”.

4.1.3. Bachillerato

En Bachillerato, se establece el siguiente objetivo [17] (artículo 11):

“g) Utilizar con solvencia y responsabilidad las tecnologías de la información y la comunicación.”

Las asignaturas que se ofertan son “Tecnologías de la Información y la Comunicación I” (pág. 533) y “Tecnologías de la Información y la Comunicación II” (pág. 534) en primer y segundo curso, respectivamente. Son asignaturas específicas, cuya oferta depende de cada Administración Autonómica y de cada centro educativo. El contenido de ambas asignaturas es de competencia digital, salvo en su contenido de programación.

Puede añadirse la asignatura “Tecnología Industrial II”, de segundo de Bachillerato. También es una asignatura específica de oferta dependiente de cada Administración Autonómica y de cada centro educativo. Consta de 5 bloques, de los que los bloques cuarto y quinto son, respectivamente, “Circuitos y sistemas lógicos” y “Control y programación de sistemas automáticos” (que principalmente incluye circuitos secuenciales).

Esta escasa presencia de la informática también se refleja en que no forma parte de las pruebas de acceso a la Universidad, cuando hay bastantes grados que podrían ponderar dicha prueba por los contenidos de varias de sus asignaturas.

4.1.4. Legislación autonómica

La legislación educativa en España es compleja, con un reparto de competencias en la planificación y desarrollo de las asignaturas entre gobierno central, gobiernos autonómicos y centros educativos [16][17]. En el ejercicio de sus competencias, las comunidades autonómicas pueden establecer o complementar contenidos, criterios de evaluación o estándares de aprendizaje evaluables para las asignaturas, según el caso. También pueden definir asignaturas de libre configuración autonómica.

Existe una gran diversidad de situaciones. Según informes elaborados por el INTEF [39][40], la mayoría de las comunidades autonómicas han introducido algunas asignaturas sobre programación, robótica o pensamiento computacional en Educación Secundaria. Destacan algunas comunidades donde su oferta es obligatoria, como Navarra en Educación Primaria, la Comunidad Valenciana en Educación Secundaria, y Madrid y Cataluña en ambas etapas educativas. También tiene relevancia para la propuesta presentada la propuesta normativa sobre Competencia Digital del Alumnado presentada por el Gobierno Vasco [13], porque detalla las competencias a alcanzar al final de Educación Primaria y Educación Secundaria Obligatoria.

4.2. Otras iniciativas

Son numerosas las iniciativas realizadas con carácter no oficial desde numerosas asociaciones, empresas, colectivos e individuos. Puede encontrarse una relación no exhaustiva en [11][15][32]. Cabe destacar las recomendaciones realizadas por asociaciones de profesores de informática en Educación Secundaria a partir de su experiencia docente, como la Asociación Andaluza de Profesores de Informática (AAPRI) [3], la Asociación de Profesores de Informática de Castilla-La Mancha (ARPICM) [4] y la Asociación de Profesores de Informática de la Comunidad Valenciana (APICV) [5]. También se encuentran algunas recomendaciones efectuadas individualmente por algunos profesores (p.ej. [29]).

5. Propuesta de organización de la educación en informática

En esta sección presentamos una propuesta de organización de las enseñanzas conducente al aprendizaje de la competencia digital y la informática.

5.1. Planteamiento global

Dada la gran importancia de la competencia digital y de la informática, los alumnos deben educarse en ambos aspectos. La única manera de garantizar una formación universal en el sistema educativo español es mediante una materia, troncal o específica, de obligado estudio por parte de los alumnos.

El desarrollo de estas competencias exige una materia obligatoria independiente de otras materias, incluso de materias relacionadas pero con contenidos distintos, como Matemáticas o Tecnología. De hecho, las tecnologías industrial e informática tienen un carácter muy distinto, el cual se manifiesta, por ejemplo, en basarse en matemáticas diferentes o en el distinto perfil de su profesorado.

Llamaremos “Informática” a esta nueva materia, que tendrá como objetivo el aprendizaje de la competencia digital y de la ciencia y técnica informática. Consideramos preferible el agrupamiento en una sola materia de estos contenidos, dada la clara relación entre ambos. Es lo mismo que sucede con la materia de “Lengua”, que agrupa su estudio (léxico, sintaxis, etc.) y su ejercicio (lectura, escritura, etc.).

La materia “Informática” debe concretarse en una asignatura obligatoria que se ofrezca en todos los cursos, desde Educación Primaria hasta Bachillerato pasando por la Educación Secundaria Obligatoria. Esta materia debe reglarse legislativamente de forma que se detallen sus contenidos, criterios de evaluación y estándares de aprendizaje evaluables, como en el resto de materias.

La existencia de una materia obligatoria donde los alumnos puedan adquirir la competencia digital facilitaría el uso instrumental de la tecnología digital en otras asignaturas, como sucede con las matemáticas o la lengua.

En los apartados siguientes desarrollamos una propuesta de objetivos de aprendizaje, que se ha realizado teniendo en cuenta los siguientes criterios:

- Objetivos claros. Se incluyen contenidos factuales, conceptuales y procedimentales orientados a la adquisición de la competencia digital y a un conocimiento básico de los fundamentos de la informática. Dicha claridad de objetivos debería facilitar la planificación por parte de centros y profesorado y la evaluación de los alumnos.
- Objetivos universales. Los objetivos deben ser razonablemente alcanzables por todos los alumnos.
- Objetivos adecuados a cada etapa educativa. Hemos tenido en cuenta tres factores importantes: la limitada capacidad de autoconciencia y de abstracción de los alumnos en estas etapas, priorizar conocimientos que sean susceptibles de práctica (evitando la simple memorización), y restricciones de uso de diversos medios digitales por razones de edad.

Los contenidos de competencia digital y de ciencia informática deben variar según las etapas educativas y el correspondiente grado de madurez de los alumnos:

- Educación Primaria. Se incluyen contenidos de competencia digital y conocimientos básicos de programación, computadores, redes y datos. La competencia digital asegura unos conocimientos y habilidades básicos para el uso eficaz, seguro y ético de las tecnologías de la información.
- Educación Secundaria Obligatoria. Es similar, pero los conocimientos, sobre todo de programación, adquieren un carácter más abstracto. La competencia digital se desarrolla a un nivel de usuario medio.
- Bachillerato. Se distingue entre las modalidades de Ciencias, Ciencias Sociales y Humanidades, y Artes. En todos los casos se profundiza en conocimientos de ciencia informática y se resalta la aplicación de herramientas informáticas en su ámbito.

5.2. Áreas y objetivos

La materia Informática consta de seis áreas:

1. **Programación.** Incluye conceptos, lenguajes y actividades de programación (p.ej., “bucle”, “lenguaje visual” y “depurar”, respectivamente), así como otros conocimientos relacionados (p.ej. el razonamiento lógico y los algoritmos).
2. **Ordenadores y sistemas operativos.** Incluye el ordenador y otros dispositivos digitales, sus componentes, organización y funcionamiento, así como el manejo y funciones principales de los sistemas operativos.
3. **Redes e Internet.** Incluye las características, componentes, organización y funcionamiento de las redes, especialmente Internet, así como su uso.
4. **Datos.** Incluye la representación y manipulación de datos en la memoria del ordenador, en ficheros y bases de datos.
5. **Contenido digital y cooperación.** Incluye el manejo de programas de un amplio espectro y orientados a la generación de contenidos o a la comunicación.
6. **Seguridad.** Incluye los conceptos y prácticas fundamentales para el manejo seguro y ético de los dispositivos digitales.

La distinción entre las áreas no siempre es nítida. Las áreas 1-4 son áreas con una caracterización clara desde un punto de vista informático, mientras que las áreas 5-6 se han definido en buena medida para recoger objetivos de competencia digital, aunque tengan relación con algunas de las otras áreas. Por tanto, algunos de los objetivos presentados a continuación podrían haberse situado en otra área. En todo caso, consideramos más importante la identificación del objetivo que su clasificación unívoca.

La Tabla 1 presenta un resumen de los objetivos de aprendizaje que deben alcanzar los alumnos al final de cada etapa educativa. Las columnas contienen, de izquierda a derecha, los objetivos correspondientes a las etapas de Educación Primaria, Educación Secundaria Obligatoria y Bachillerato. La columna de Bachillerato recoge la modalidad de Ciencias, al ser la que desarrolla contenidos científicos y técnicos más avanzados de informática.

Conviene observar que los objetivos se han especificado utilizando un formato similar al de la taxonomía revisada de Bloom [9]. La legislación española especifica los objetivos mediante contenidos, criterios de evaluación y estándares de evaluación, pero el desarrollo de estas tres clases de especificación puede realizarse a partir de los objetivos.

Los contenidos de competencia digital y de informática se distinguen en la tabla mediante el convenio de que los primeros aparecen marcados en fuente azul y al comienzo de cada celda. Ninguna de estas diferencias de especificación tiene connotación ni relevancia educativa alguna. Cada objetivo de competencia digital tiene una etiqueta final con el formato (*m.n*). Esta etiqueta facilita conocer su relación con el marco DIGCOMP, indicando que dicho objetivo corresponde al área competencial *m* (dimensión 1 del marco) y competencia *n* (dimensión 2).

En la Tabla 2 se comparan los objetivos de las tres modalidades de Bachillerato. Los objetivos comunes a las modalidades se han colocado al comienzo de cada fila, para facilitar la identificación de coincidencias y diferencias.

En el Anexo A se incluye un análisis de la correspondencia de los objetivos de competencia digital con el marco DIGCOMP. Obsérvese que el nivel de competencia aquí especificado se alcanzaría dentro de una asignatura de Informática, pero podría desarrollarse de forma transversal con otras materias.

Tabla 1. Objetivos de aprendizaje que debe alcanzar el alumnado al final de cada etapa educativa

Área	Primaria	ESO	Bachillerato (Ciencias)
Programación	<ul style="list-style-type: none"> - Puede definir de forma ordenada y precisa los pasos necesarios para resolver un problema sencillo - Descompone un problema en problemas más sencillos que faciliten su programación - Comprende que un programa se edita y se ejecuta en un dispositivo digital usando un lenguaje de programación con instrucciones precisas y no ambiguas - Usa un lenguaje de programación visual de bloques - Construye programas que realizan tareas específicas usando secuencia, iteración, condicional, paralelismo, variables y expresiones - Usa el razonamiento lógico para predecir el comportamiento y detectar y corregir errores en programas con bloques 	<ul style="list-style-type: none"> - Usa los elementos básicos de un lenguaje de programación textual, incluyendo tipos de datos y funciones o procedimientos - Desarrolla programas para alguna plataforma distinta de los ordenadores de mesa (p.ej. web o móvil) - Comprende los conceptos básicos de los lenguajes de programación (p.ej. léxico, sintaxis, semántica y máquina virtual) - Comprende y usa la lógica booleana - Usa el razonamiento lógico para predecir el funcionamiento y detectar y corregir errores en programas textuales - Comprende las etapas del proceso de programación: análisis, diseño, edición, compilación, pruebas, depuración, documentación, desarrollo incremental - Usa un proceso de diseño iterativo para desarrollar programas - Explica el diseño de programas de forma organizada y documenta programas - Aplica casos de prueba, ya elaborados, para comprobar el correcto funcionamiento de programas implementados 	<ul style="list-style-type: none"> - Comprende algunas bases lingüísticas de los lenguajes de programación y su relación con ellos (p.ej. alfabetos y lenguajes, lenguajes regulares e independientes del contexto) - Reproduce el comportamiento de programas recursivos - Conoce estructuras de datos lineales (p.ej. pilas y colas) - Comprende algunos algoritmos fundamentales, matemáticos (p.ej. numéricos o sobre matrices) e informáticos (p.ej. ordenar o buscar) - Usa diversos criterios para evaluar algoritmos de forma sencilla (rendimiento, estilo, etc.) - Desarrolla y usa casos de prueba para algoritmos sencillos - Desarrolla programas de forma sistemática, siguiendo un ciclo de vida sencillo (análisis, diseño, etc.) - Desarrolla programas de complejidad media para diversas plataformas (p.ej. programas para ordenadores de mesa, páginas web dinámicas o apps móviles) - Desarrolla programas para campos de aplicación afines a la modalidad del bachillerato (p.ej. análisis de datos para ciencias o ciencias sociales, desarrollo artístico para artes), orientándolos a aplicaciones de actualidad (robótica, inteligencia artificial, realidad virtual o aumentada, datos masivos, etc.)
Ordenadores y sistemas operativos	<ul style="list-style-type: none"> - Sabe resolver problemas leves de manejo de dispositivos digitales y aplicaciones (5.1) - Sabe personalizar un sistema operativo (5.2) - Conoce la diferencia entre distintos dispositivos digitales (p.ej. ordenador, teléfono móvil, etc.) y entre hardware y software - Conoce los componentes de un ordenador y el papel que juegan en su funcionamiento - Comprende el papel del sistema operativo en un dispositivo informático - Maneja las unidades y múltiplos de capacidad de almacenamiento y de velocidad de procesamiento 	<ul style="list-style-type: none"> - Conoce las características y prestaciones básicas de un dispositivo digital, seleccionando el dispositivo o sus funciones más adecuadas para ciertas tareas (5.2) - Diseña circuitos digitales sencillos con puertas lógicas - Comprende los componentes hardware y software que forman un ordenador, y cómo se comunican entre sí y con otros ordenadores - Conoce las funciones principales de los sistemas operativos - Comprende cómo se almacenan y ejecutan las instrucciones en un ordenador 	<ul style="list-style-type: none"> - Utiliza máquinas virtuales para instalar y configurar un sistema operativo (5.2) - Comprende la arquitectura de von Neumann (p.ej. los elementos hardware que participan en la ejecución de instrucciones de máquina) - Comprende las tareas principales desempeñadas por un sistema operativo (p.ej. gestión de procesos o de memoria)

Redes e Internet	<ul style="list-style-type: none"> - Comprende que los ordenadores se comunican usando redes de ordenadores, incluyendo Internet - Comprende los conceptos básicos relacionados con el uso de aplicaciones basadas en redes (p.ej. navegador, buscador, URL, etc.) 	<ul style="list-style-type: none"> - Entiende los distintos tipos de redes, el papel de las mismas en Internet y las principales tecnologías de cada tipo (p.ej. Ethernet y wifi; fibra óptica, telefonía fija, telefonía móvil y satélite) - Conoce las características y la función de los dispositivos que forman una red local (p.ej. tarjetas de red, conmutadores, enrutadores, puntos de acceso wifi o repetidores wifi) - Conoce los conceptos y elementos necesarios para comprender el funcionamiento de las redes de computadores (p.ej. servidor, cliente, dirección IP, sistema de nombres de dominio, etc.) 	<ul style="list-style-type: none"> - Configura y usa con seguridad redes inalámbricas domésticas, e interconecta y configura dispositivos (4.1, 5.2) - Conoce las características principales de algunos protocolos (p.ej. protocolos de manejo del correo electrónico o de enrutamiento) - Configura y usa algún servicio remoto (p.ej. FTP)
Datos	<ul style="list-style-type: none"> - Sabe almacenar y recuperar ficheros (1.3) - Conoce el sistema de directorios y sabe navegar por él (1.3) - Comprende cómo se representan los números en el sistema binario y realiza operaciones sencillas con números binarios - Comprende que los ordenadores procesan unos datos (de entrada) para generar otros (de salida) 	<ul style="list-style-type: none"> - Recopila y procesa datos, visualiza y valora sus resultados (p.ej. con una hoja de cálculo) (1.2, 3.2) - Sabe organizar la información en ficheros y directorios con nombres adecuados (1.3) - Comprende las implicaciones de los diferentes formatos de almacenamiento en la calidad y el tamaño de los datos (5.2) - Convierte entre distintos sistemas de numeración (al menos, binarios, decimales y hexadecimales) - Comprende cómo se representan y manipulan, en forma de dígitos binarios, datos de varios tipos (p.ej. texto, sonidos y gráficos) 	<ul style="list-style-type: none"> - Realiza consultas básicas a bases de datos con el lenguaje SQL (1.1, 1.2, 1.3) - Administra de forma básica un gestor de base de datos (p.ej. crear una base de datos, crear formularios e informes, etc.) (1.1, 1.2, 1.3) - Conoce algunos principios del análisis de datos masivos para extraer conocimiento
Contenido digital y colaboración	<ul style="list-style-type: none"> - Conoce la terminología básica relacionada con el manejo de dispositivos digitales y programas (p.ej. ratón, ventana, etc.) y lo realiza - Realiza las principales clases de interacción (p.ej. seleccionar, copiar, pegar, etc.) con distintos dispositivos de entrada/salida (teclado, ratón e interfaces multimodales) - Sabe navegar y buscar información en la web (1.1) - Valora si información encontrada en la web es relevante (1.2) - Utiliza plataformas y herramientas de comunicación y colaboración del centro (2.1, 2.2, 2.4) - Usa los programas más comunes de contenido digital a un nivel básico (procesamiento de textos, hojas de cálculo, presentaciones, páginas web, y manejo de imágenes, gráficos, vídeo y material multimedia) (3.1) 	<ul style="list-style-type: none"> - Hace búsquedas compuestas y filtra información en la web (1.1) - Valora si la información encontrada en la web es de calidad (1.2) - Sabe manejar y organizar el correo electrónico (2.1) - Sabe compartir información en Internet y usarla para colaborar (p.ej. almacenamiento o edición en la nube) (2.2, 2.4) - Realiza alguna gestión online en el centro educativo (2.3) - Sabe crear y manejar su identidad digital (2.6) - Usa los programas más comunes de contenido digital a un nivel intermedio (procesamiento de textos, hojas de cálculo, presentaciones, páginas web, y manejo de imágenes, gráficos, vídeo y material multimedia) (3.1, 3.2) 	<ul style="list-style-type: none"> - Conoce y usa otras herramientas de comunicación de Internet (p.ej. redes sociales, mensajería instantánea, blogs, foros, telefonía IP, videoconferencia, etc.) (2.1, 2.2, 2.4) - Usa programas de contenido digital relevantes en la modalidad a un nivel avanzado (p.ej. hojas de cálculo, diseño 2D y 3D o cartografía digital en ciencias, hojas de cálculo en ciencias sociales, multimedia en artes) (3.1, 3.2) - Gestiona adecuadamente los derechos de autor de contenidos digitales propios o ajenos (3.3) - Sabe seleccionar la herramienta de comunicación o de creación de contenido más adecuada para el fin previsto (5.2)

	<ul style="list-style-type: none"> - Cita las fuentes de las que ha obtenido la información (3.3) 	<ul style="list-style-type: none"> - Tiene conocimientos de derechos de autor y licencias, incluyendo software libre (3.3) - Sabe resolver problemas leves de manejo de aplicaciones (5.1) - Configura programas a sus gustos o necesidades (5.2) 	
Seguridad	<ul style="list-style-type: none"> - Conoce unas normas básicas de comportamiento virtual (2.5) - Es consciente de comportamientos digitales no éticos (p.ej. acoso o identidad falsa) (2.5) - Sabe cómo manejar dispositivos digitales y contenidos digitales de forma segura (4.1) - Conoce formas básicas de protección de datos (p.ej. copias de seguridad) y de la intimidad (p.ej. contraseñas) (4.2) - Es consciente de los riesgos de Internet para la intimidad y la salud física (p.ej. malas posturas) y mental (p.ej. acoso o adicción) (4.3) 	<ul style="list-style-type: none"> - Sabe manejar comportamientos digitales no seguros (p.ej. transacciones inseguras) o no éticos (2.5, 4.2) - Sabe proteger dispositivos y contenidos digitales (p.ej. antivirus y copias de seguridad) y datos personales (p.ej. contraseñas fuertes) (4.1, 4.2) - Sabe evitar los riesgos del manejo de dispositivos físicos para la salud física, mental o el medio ambiente (p.ej. material informático desechado) (4.3, 4.4) - Comprende cómo funcionan los mecanismos de detección y corrección de errores (p.ej. bits de paridad). 	<ul style="list-style-type: none"> - Comprende los principios de seguridad de la información (confidencialidad, integridad, disponibilidad) - Tiene conocimientos básicos de criptografía

Tabla 2. Objetivos de aprendizaje para las tres modalidades de Bachillerato

Área	Ciencias	Ciencias Sociales y Humanidades	Arte
Programación	<ul style="list-style-type: none"> - Comprende algunas bases lingüísticas de los lenguajes de programación y su relación con ellos (p.ej. alfabetos y lenguajes, lenguajes regulares e independientes del contexto) - Conoce estructuras de datos lineales (p.ej. pilas y colas) - Comprende algunos algoritmos fundamentales, matemáticos (p.ej. numéricos o sobre matrices) e informáticos (p.ej. ordenar o buscar) - Desarrolla y usa casos de prueba para algoritmos sencillos - Desarrolla programas de forma sistemática, siguiendo un ciclo de vida sencillo (análisis, diseño, etc.) - Desarrolla programas para campos de aplicación afines a la modalidad de Ciencias (p.ej. análisis de datos), orientándolos a aplicaciones de actualidad (robótica, inteligencia artificial, realidad virtual o aumentada, datos masivos, etc.) - Desarrolla programas de complejidad media para diversas plataformas (p.ej. programas para ordenadores de mesa, páginas web dinámicas o apps móviles) - Reproduce el comportamiento de programas recursivos - Usa diversos criterios para evaluar algoritmos de forma sencilla (rendimiento, estilo, etc.) 	<ul style="list-style-type: none"> - Comprende algunas bases lingüísticas de los lenguajes de programación y su relación con ellos (p.ej. alfabetos y lenguajes, lenguajes regulares e independientes del contexto) - Conoce estructuras de datos lineales (p.ej. pilas y colas) - Comprende algunos algoritmos fundamentales, matemáticos (p.ej. numéricos o sobre matrices) e informáticos (p.ej. ordenar o buscar) - Desarrolla y usa casos de prueba para algoritmos sencillos - Desarrolla programas de forma sistemática, siguiendo un ciclo de vida sencillo (análisis, diseño, etc.) - Desarrolla programas para campos de aplicación afines a la modalidad de Ciencias Sociales y Humanidades (p.ej. análisis de datos), orientándolos a aplicaciones de actualidad (robótica, inteligencia artificial, realidad virtual o aumentada, datos masivos, etc.) 	<ul style="list-style-type: none"> - Comprende algunas bases lingüísticas de los lenguajes de programación y su relación con ellos (p.ej. alfabetos y lenguajes, lenguajes regulares e independientes del contexto) - Conoce estructuras de datos lineales (p.ej. pilas y colas) - Comprende algunos algoritmos fundamentales, matemáticos (p.ej. numéricos o sobre matrices) e informáticos (p.ej. ordenar o buscar) - Desarrolla y usa casos de prueba para algoritmos sencillos - Desarrolla programas de forma sistemática, siguiendo un ciclo de vida sencillo (análisis, diseño, etc.) - Desarrolla programas para campos de aplicación afines a la modalidad de Artes (p.ej. desarrollo artístico), orientándolos a aplicaciones de actualidad (robótica, inteligencia artificial, realidad virtual o aumentada, datos masivos, etc.)
Ordenadores y sistemas operativos	<ul style="list-style-type: none"> - Utiliza máquinas virtuales para instalar y configurar un sistema operativo - Comprende la arquitectura de von Neumann (p.ej. los elementos hardware que participan en la ejecución de instrucciones de máquina) - Comprende las tareas principales desempeñadas por un sistema operativo (p.ej. gestión de procesos o de memoria) 	<ul style="list-style-type: none"> - Utiliza máquinas virtuales para instalar y configurar un sistema operativo 	<ul style="list-style-type: none"> - Utiliza máquinas virtuales para instalar y configurar un sistema operativo

Redes e Internet	<ul style="list-style-type: none"> - Configura y usa con seguridad redes inalámbricas domésticas, e interconecta y configura dispositivos (incluyendo redes de sensores, dispositivos vestibles, etc.) - Configura y usa servicios remotos - Conoce las características principales de algunos protocolos (p.ej. protocolos de manejo del correo electrónico o de enrutamiento) 	<ul style="list-style-type: none"> - Configura y usa con seguridad redes inalámbricas domésticas, e interconecta y configura dispositivos (incluyendo redes de sensores, dispositivos vestibles, etc.) - Configura y usa servicios remotos 	<ul style="list-style-type: none"> - Configura y usa con seguridad redes inalámbricas domésticas, e interconecta y configura dispositivos (incluyendo redes de sensores, dispositivos vestibles, etc.) - Configura y usa servicios remotos
Datos	<ul style="list-style-type: none"> - Realiza consultas básicas a bases de datos con el lenguaje SQL - Conoce algunos principios del análisis de datos masivos para extraer conocimiento - Administra de forma básica un gestor de base de datos (p.ej. crear una base de datos, crear formularios e informes, etc.) 	<ul style="list-style-type: none"> - Realiza consultas básicas a bases de datos con el lenguaje SQL - Conoce algunos principios del análisis de datos masivos para extraer conocimiento - Diseña bases de datos sencillas, creando consultas, formularios e informes 	<ul style="list-style-type: none"> - Realiza consultas básicas a bases de datos con el lenguaje SQL - Conoce algunos principios del análisis de datos masivos para extraer conocimiento - Diseña bases de datos sencillas, creando consultas, formularios e informes
Contenido digital y comunicación	<ul style="list-style-type: none"> - Usa programas de contenido digital a un nivel avanzado (p.ej. hojas de cálculo, diseño 2D y 3D, imagen digital, etc.) - Sabe crear y usar medios de comunicación sociales (p.ej. blogs) 	<ul style="list-style-type: none"> - Usa programas de contenido digital a un nivel avanzado (p.ej. hojas de cálculo, paquetes estadísticos, etc.) - Sabe crear y usar medios de comunicación sociales (p.ej. blogs) - Comprende y aplica técnicas de marketing y comercio electrónico en el uso y aprovechamiento de redes sociales 	<ul style="list-style-type: none"> - Usa programas de contenido digital a un nivel avanzado (p.ej. diseño 2D y 3D, imagen digital, multimedia, etc.) - Sabe crear y usar medios de comunicación sociales (p.ej. canales de distribución multimedia) - Comprende y aplica técnicas de marketing y comercio electrónico en el uso y aprovechamiento de redes sociales
Seguridad	<ul style="list-style-type: none"> - Comprende los principios de seguridad de la información (confidencialidad, integridad, disponibilidad) - Tiene conocimientos básicos de criptografía 	<ul style="list-style-type: none"> - Comprende los principios de seguridad de la información (confidencialidad, integridad, disponibilidad) 	<ul style="list-style-type: none"> - Comprende los principios de seguridad de la información (confidencialidad, integridad, disponibilidad)

6. Implantación

En esta sección tratamos brevemente dos cuestiones clave para una implantación exitosa de las propuestas: su didáctica y la formación del profesorado.

6.1. Didáctica

Una cuestión importante es el uso de una didáctica cuidada para la nueva materia de Informática (análogamente a otras materias, como Matemáticas). Esta didáctica no existe actualmente, pero se pueden señalar algunas líneas necesarias para su desarrollo:

- Definición de itinerarios y niveles de competencias que permitan organizar las enseñanzas en las sucesivas etapas educativas. Esto implicará determinar el nivel cognitivo exigido al alumnado, la introducción precisa de conceptos, su sucesiva ejercitación rutinaria y (relativamente) creativa, y su repetición en cursos sucesivos. Aunque se parte de una propuesta razonable, debe investigarse su adecuación a cada edad.
- Disponibilidad de libros de texto y materiales docentes de calidad. Un libro de texto es necesario para que el profesor tenga una guía, que el alumno pueda resolver dudas y estudiar por su cuenta, y que los padres puedan ayudar a sus hijos.
- Lectura de programas de calidad. La lectura de programas de calidad es una forma de aprender. Algunos entornos, como Scratch, favorecen este hábito mediante la compartición del código con una comunidad. Incluso se permite modificar el código y volver a compartir, con reconocimiento a sus autores originales.
- Herramientas hardware y software adecuadas a cada edad. Las herramientas de programación disponibles permiten desarrollar y ejecutar programas con facilidad, pero suelen ser claramente mejorables. Con frecuencia se echan en falta herramientas auxiliares pero necesarias, como los depuradores.
- Ejercicios adecuados:
 - Variedad de clases de ejercicios. El aprendizaje basado en proyectos es adecuado para un nivel medio o avanzado en informática pero no para los niveles iniciales, en los cuales son preferibles otro tipo de ejercicios y prácticas más sencillas. Es un error que se ha cometido en las asignaturas de introducción a la programación en la universidad [43] y que no debería repetirse en las etapas educativas previas. Estos proyectos tradicionales están orientados al nivel “Crear” de la taxonomía revisada de Bloom [9], pero los alumnos principiantes necesitan alcanzar previamente niveles inferiores de aprendizaje. El Anexo B contiene una relación de ejercicios utilizados con éxito en la universidad y que pueden adaptarse o servir de inspiración en estas etapas.
 - El trabajo en equipo, y especialmente el trabajo en parejas, se ha mostrado como una buena práctica de aprendizaje de la programación. Existen numerosas experiencias a nivel universitario y profesional [35], que pueden adaptarse para edades inferiores.

- Evitar los enunciados basados en elementos de lenguajes. Este tipo de enunciados deja libertad al alumno para desarrollar un programa cualquiera, exigiéndole como único requisito que utilice ciertos elementos del lenguaje de programación (p.ej. bucles condicionales). Se trata de un enunciado artificioso, donde el alumno frecuentemente se ve forzado a introducir elementos de forma innecesaria. Es preferible que el profesor especifique el comportamiento esperado del programa y que lo haya resuelto anteriormente, comprobando qué elementos de lenguaje son convenientes o necesarios.
- Evitar los ejercicios desenfocados. En algunos ejercicios, el alumno debe dedicar excesivo tiempo a cuestiones rutinarias (por ejemplo, programas Java donde el alumno deba manejar ficheros para la entrada y salida de datos). Esta clase de ejercicios es especialmente desafortunada si las actividades rutinarias se repiten continuamente porque aburren y desmotivan a los alumnos.

6.2. Formación del profesorado

Un elemento clave para una adecuada educación en informática de los alumnos es la formación del profesorado. No es necesaria la misma formación para impartir competencia digital que informática y programación. Para lo primero, basta con que el profesorado tenga competencia digital docente certificada; para lo segundo, debe tener una adecuada formación en informática.

La situación es distinta en Educación Primaria que en las etapas posteriores. El profesorado de Educación Primaria tiene un perfil genérico y puede impartir cualquier materia (con algunas excepciones). Dado que un porcentaje alto de las competencias a desarrollar en Educación Primaria son de competencia digital, los profesores de Educación Primaria necesitan obtener un certificado en competencia digital docente. Afortunadamente, el Ministerio de Educación, por medio del INTEF, ha desarrollado un marco de competencia digital docente [38] y su evaluación mediante un portafolio. Por otro lado, sería deseable que los Grados y Posgrados en Educación Primaria se modificaran para proporcionar una sólida competencia digital a los futuros profesores. De esta forma, este profesorado no sólo podría impartir el contenido en competencia digital de la asignatura de Informática sino que también podrían utilizarla transversalmente en otras asignaturas. Además, sus estudios de Grado y Posgrado deberían modificarse para permitirles adquirir unos conocimientos mínimos de informática y programación.

La asignatura de Informática en Educación Secundaria Obligatoria o Bachillerato tiene un alto contenido de informática. Por tanto, es necesario que el profesorado tenga un perfil técnico en informática (o afín) con formación docente. La forma de garantizar esta formación es modificar los Másteres en Formación del Profesorado, desglosando el perfil “Informática y Tecnología” existente en algunas ofertas en dos perfiles claramente diferenciados.

Anexo A. Correspondencia de los objetivos de competencia digital con el marco DIGCOMP

Presentamos la correspondencia de los objetivos educativos de competencia digital con el marco DIGCOMP. En la Tabla 3 se presentan los objetivos agrupados por áreas competenciales de DIGCOMP (dimensión 1). Cada objetivo se etiqueta al final con su área competencial y su competencia (dimensiones 1 y 2) correspondientes.

En la Tabla 4 se realiza un análisis más general, indicando el nivel de maestría esperado para cada competencia. Puede observarse que la mayor parte de las competencias deben desarrollarse a nivel básico, intermedio y avanzado según se avanza por las etapas de Educación Primaria, Educación Secundaria Obligatoria y Bachillerato. En todas estas competencias es posible encontrar conceptos y tareas de complejidad creciente desde un punto de vista informático.

Pueden extraerse varias conclusiones sobre el marco DIGCOMP tras analizar dicha correspondencia. En líneas generales, constituye un marco bien estructurado que resulta útil para la organización de objetivos educativos. Además, la mayor parte de las competencias pueden desarrollarse en todas las etapas educativas, permitiendo alcanzar en Bachillerato un nivel avanzado de maestría.

Sin embargo, el marco está concebido para el ciudadano, no para la educación del niño o del joven. Por tanto, hay que tener la precaución de no plasmarlo directamente en la legislación educativa, sino que debe adaptarse a cada etapa educativa. Veamos algunas carencias:

- El ámbito de las competencias en DIGCOMP es muy variado, desde algunas genéricas hasta otras específicas. El desarrollo de algunas competencias exige un alto número de conceptos, prácticas y tiempo (por ejemplo, el desarrollo de contenido digital – competencia 3.1–, ya que hay muchas clases de contenidos digitales relevantes), mientras que el desarrollo de otras será más superficial (por ejemplo, la protección del medio ambiente –4.4–).
- Los niveles iniciales de competencia digital no se contemplan en ningún área competencial de DIGCOMP (al menos, de forma evidente). Estos primeros niveles deben ser alcanzados por cualquier persona que se inicie en el manejo de dispositivos digitales y aplicaciones, y obviamente también por los niños.
- El marco incluye competencias que son más adecuadas para un adulto (como identificar lagunas de competencia digital –5.4–) o incluso para un técnico (la resolución de problemas técnicos –5.4–), sobre todo a un nivel avanzado.

Tabla 3. Correspondencia entre objetivos de aprendizaje de competencia digital y competencias de DIGCOMP

Dimensión	Primaria	ESO	Bachillerato (Ciencias)
Manejo básico de dispositivos digitales	<ul style="list-style-type: none"> – Conoce la terminología básica relacionada con el manejo de dispositivos digitales y programas (p.ej. ratón, ventana, etc.) y lo realiza – Realiza las principales clases de interacción (p.ej. seleccionar, copiar, pegar, etc.) con distintos dispositivos de entrada/salida (teclado, ratón e interfaces multimodales) 		
1. Alfabetización en información y datos	<ul style="list-style-type: none"> – Sabe navegar y buscar información en la web (1.1) – Valora si la información encontrada en la web es relevante (1.2) – Sabe almacenar y recuperar ficheros (1.3) – Conoce el sistema de directorios y sabe navegar por él (1.3) 	<ul style="list-style-type: none"> – Hace búsquedas compuestas y filtra información en la web (1.1) – Valora si información encontrada en la web es de calidad (1.2) – Recopila y procesa datos, visualiza y valora sus resultados (p.ej. con una hoja de cálculo) (1.2, 3.2) – Sabe organizar la información en ficheros y directorios con nombres adecuados (1.3) 	<ul style="list-style-type: none"> – Realiza consultas básicas a bases de datos con el lenguaje SQL (1.1, 1.2, 1.3) – Administra de forma básica un gestor de base de datos (p.ej. crear una base de datos, crear formularios e informes, etc.) (1.1, 1.2, 1.3)
2. Comunicación y colaboración	<ul style="list-style-type: none"> – Utiliza plataformas y herramientas de comunicación y colaboración del centro (2.1, 2.2, 2.4) – Conoce unas normas básicas de comportamiento virtual (2.5) – Es consciente de comportamientos digitales no éticos (p.ej. acoso o identidad falsa) (2.5) 	<ul style="list-style-type: none"> – Sabe manejar y organizar el correo electrónico (2.1) – Sabe compartir información en Internet y usarla para colaborar (p.ej. almacenamiento o edición en la nube) (2.2, 2.4) – Realiza alguna gestión online en el centro educativo (2.3) – Sabe manejar comportamientos digitales no seguros (p.ej. transacciones inseguras) o no éticos (2.5, 4.2) – Sabe crear y manejar su identidad digital (2.6) 	<ul style="list-style-type: none"> – Conoce y usa otras herramientas de comunicación de Internet (p.ej. redes sociales, mensajería instantánea, blogs, foros, telefonía IP, videoconferencia, etc.) (2.1, 2.2, 2.4)
3. Creación de contenido digital	<ul style="list-style-type: none"> – Usa los programas más comunes de contenido digital a un nivel básico (procesamiento de textos, hojas de cálculo, presentaciones, páginas web, y manejo de imágenes, gráficos, vídeo y material multimedia) (3.1, 3.2) – Cita las fuentes de las que ha obtenido la información (3.3) 	<ul style="list-style-type: none"> – Usa los programas más comunes de contenido digital a un nivel intermedio (procesamiento de textos, hojas de cálculo, presentaciones, páginas web, y manejo de imágenes, gráficos, vídeo y material multimedia) (3.1, 3.2) – Tiene conocimientos de derechos de autor y licencias, incluyendo software libre (3.3) 	<ul style="list-style-type: none"> – Usa programas de contenido digital relevantes en la modalidad a un nivel avanzado (p.ej. hojas de cálculo, diseño 2D y 3D o cartografía digital en ciencias, hojas de cálculo en ciencias sociales, multimedia en artes) (3.1, 3.2) – Gestiona adecuadamente los derechos de autor de contenidos digitales propios o ajenos (3.3)
4. Seguridad	<ul style="list-style-type: none"> – Sabe cómo manejar dispositivos digitales y contenidos digitales de forma segura (4.1) – Conoce formas básicas de protección de datos (p.ej. copias de seguridad) y de la intimidad (p.ej. contraseñas) (4.2) 	<ul style="list-style-type: none"> – Sabe proteger dispositivos y contenidos digitales (p.ej. antivirus y copias de seguridad) y datos personales (p.ej. contraseñas fuertes) (4.1, 4.2) – Sabe evitar los riesgos del manejo de dispositivos físicos para la salud física, mental o el medio 	<ul style="list-style-type: none"> – Configura y usa con seguridad redes inalámbricas domésticas, e interconecta y configura dispositivos (4.1, 5.2)

	<ul style="list-style-type: none"> - Es consciente de los riesgos de Internet para la intimidad y la salud física (p.ej. malas posturas) y mental (p.ej. acoso o adicción) (4.3) 	ambiente (p.ej. material informático desechado) (4.3, 4.4)	
5. Problemas técnicos	<ul style="list-style-type: none"> - Sabe resolver problemas leves de manejo de dispositivos digitales y aplicaciones (5.1) - Sabe personalizar un sistema operativo (5.2) 	<ul style="list-style-type: none"> - Configura programas a sus gustos o necesidades (5.2) - Comprende las implicaciones de los diferentes formatos de almacenamiento en la calidad y el tamaño de los datos (5.2) - Conoce las características y prestaciones básicas de un dispositivo digital o sus aplicaciones, seleccionando las características más adecuadas para ciertas tareas (5.2) 	<ul style="list-style-type: none"> - Utiliza máquinas virtuales para instalar y configurar un sistema operativo (5.2) - Sabe seleccionar la herramienta de comunicación o de creación de contenido más adecuada para el fin previsto (5.2)

Tabla 4. Clasificación del desarrollo de las competencias digitales propuestas en niveles de maestría según su correspondencia con DIGCOMP

Competencia	Primaria	ESO	Bachillerato (Ciencias)
1.1. Navegar, buscar y filtrar datos, información y contenido digital	Básico	Intermedio	Avanzado
1.2. Evaluar datos, información y contenido digital	Básico	Intermedio	Avanzado
1.3. Manejar datos, información y contenido digital	Básico	Intermedio	Avanzando
2.1. Interactuar mediante tecnologías digitales	Básico	Intermedio	Avanzado
2.2. Compartir mediante tecnologías digitales	Básico	Intermedio	Avanzado
2.3. Participar en la ciudadanía mediante tecnologías digitales	–	Básico	–
2.4. Colaborar mediante tecnologías digitales	Básico	Intermedio	Avanzado
2.5. Comportamiento digital	Básico	Intermedio	–
2.6. Gestionar la identidad digital	–	Básico	–
3.1. Desarrollar contenido digital	Básico	Intermedio	Avanzado
3.2. Integrar y reelaborar contenido digital	Básico	Intermedio	Avanzado
3.3. Derechos de autor y licencias	Básico	Intermedio	Avanzado
3.4. Programar	Básico	Intermedio	Avanzado
4.1. Proteger dispositivos digitales	Básico	Intermedio	Avanzado
4.2. Proteger datos y la intimidad personales	Básico	Intermedio	–
4.3. Proteger la salud y el bienestar	Básico	Intermedio	–
4.4. Proteger el medio ambiente	–	Básico	–
5.1. Resolver problemas técnicos	Básico	–	–
5.2. Identificar necesidades y respuestas tecnológicas	Básico	Intermedio	Avanzado
5.3. Usar las tecnologías digitales con creatividad	–	–	–
5.4. Identificar lagunas de competencia digital	–	–	–

Anexo B. Ejercicios de programación

Veamos algunas clases de ejercicios que facilitan un aprendizaje gradual y que se han probado con éxito en la universidad. Por brevedad, solamente incluimos una pequeña muestra de tipos de ejercicios y sin ejemplos ilustrativos. Pueden encontrarse más tipos de ejercicios en algunas publicaciones [19][36][54][59].

- Ejercicios predictivos [44]. Se proporciona un fragmento de código y se pide al alumno que identifique el resultado de su ejecución. Suele preguntarse mediante tests de múltiples respuestas, donde los distractores deben tener relación con fallos o malentendidos frecuentes. La variedad de resultados a pedir es grande: valor de una expresión, valor almacenado en una variable, resultado de una función, mensaje impreso en salida estándar, número de iteraciones de un bucle, etc.
- Ejercicios de “esquema” [44]. Se especifica el comportamiento deseado y se proporciona un fragmento incompleto de código, que el alumno debe completar.
- Ejercicios de Parsons [52]. Se especifica el comportamiento deseado y un trozo de código que lo satisface, pero cuyas instrucciones están desordenadas. Se pide ordenarlas adecuadamente.
- Escribir una traza [33]. Se pide escribir una traza que ilustre el comportamiento de un trozo de programa. Existen estudios que avalan que la comprensión de trazas es un conocimiento previo, o al menos más sencillo, que la escritura de código nuevo [45].
- Se pide escribir un programa o fragmento de código que proporcione un ejemplo de algún comportamiento [19].
- Desarrollar un programa o fragmento de programa que satisfaga una especificación y que tenga una estructura dada [19].
- Depuración de programas [19]. Se proporciona un programa que no tiene el comportamiento deseado y se pide que se corrija. También puede pedirse que primero se identifiquen las razones de su comportamiento indeseado.

Referencias

- [1] ACM (2017): *Curricula Recommendations*, <https://www.acm.org/education/curricula-recommendations>.
- [2] ACM, Code.org, CSTA, Cyber Innovation Center, y National Math and Science Initiative (2016): *K-12 Computer Science Framework*, <http://www.k12cs.org>.
- [3] Asociación Andaluza de Profesores de Informática (sin fecha): *Propuesta de la AAPRI para un currículum de Informática en ESO y Bachillerato*, <http://www.aapri.es>.
- [4] Asociación de Profesores de Informática de Castilla-La Mancha (sin fecha): *Aportaciones de ARPICM a los currículos de Informática y TIC en la ESO y Bachillerato*, <https://www.arpicm.es/>.
- [5] Asociación de Profesores de Informática de la Comunidad Valenciana (sin fecha): *Propuesta para primaria y para la ESO*, <https://www.apicv.es/>.
- [6] ACM e IEEE Computing Society, the Joint Task Force on Computing Curricula (2013): *Computer Science Curricula 2013 – Curriculum Guidelines for Undergraduate Degree Programs in Computer Science*, https://www.acm.org/binaries/content/assets/education/cs2013_web_final.pdf.
- [7] A. V. Aho (2011): “Computation and computational thinking”, *Ubiquity*, DOI [10.1145/1895419.1922682](https://doi.org/10.1145/1895419.1922682).
- [8] K. Ala-Mutka (2011): *Mapping digital competence: Towards a conceptual understanding*, informe JRC67075, Institute for Prospective Technological Studies, Joint Research Centre, Comisión Europea.
- [9] L.W. Anderson, D.R. Krathwohl, P.W. Airasian, K.A. Cruikshank, R.E. Mayer, P.R. Pintrich, R. Raths y M.C. Wittrock (2001): *A Taxonomy for Learning, Teaching and Assessing – A revision of Bloom’s taxonomy of educational objectives*, Addison-Wesley Longman.
- [10] M. Armoni (2016): “Computing in schools: Computer science, computational thinking, programming, coding: The anomalies of transitivity in K-12 computer science education”, *ACM Inroads*, 7(4):24-27, DOI [10.1145/3011071](https://doi.org/10.1145/3011071).
- [11] A. Balanskat y K. Engelhardt (2015): *Computing our future: Computer programming and coding - Priorities, school curricula and initiatives across Europe*, European Schoolnet, http://fcl.eun.org/documents/10180/14689/Computing+our+future_final.pdf/746e36b1-e1a6-4bf1-8105-ea27c0d2bbe0.
- [12] E. Barendsen, L. Mannila, B. Demo, N. Grgurina, C. Izu, C. Mirolo, S. Sentance, A. Settle y G. Stupuriené (2015): “Concepts in K–9 computer science education”, *ITiCSE’15 Working Group Reports*, DOI [10.1145/2858796.2858800](https://doi.org/10.1145/2858796.2858800).
- [13] Berritzegune de Sestao, Site de Competencia en el Tratamiento de la Información y Competencia digital (2017): <https://sites.google.com/a/b03sestao.es/ikt-b03-sestao/competencia-digital>.

- [14] M. Berry (2013): *Computing in the National Curriculum – A guide for primary teachers*, Computing at School.
- [15] S. Bocconi, A. Chiocciariello, G. Dettori, A. Ferrari y K. Engelhardt (2016): *Developing Computational Thinking in Compulsory Education*, informe EUR 28295, Institute for Prospective Technological Studies, Joint Research Centre, Comisión Europea.
- [16] BOE (2014): Real Decreto 126/2014, de 18 de febrero, por el que se establece el currículo básico de la Educación Primaria, 1 marzo 2014, nº 52, sec. I, pp. 19.349-19.420.
- [17] BOE (2015): Real Decreto 1.105/2014, de 26 de diciembre, por el que se establece el currículo básico de la Educación Secundaria Obligatoria y del Bachillerato, 3 enero 2015, nº 3, sec. I, pp. 169-546.
- [18] BOE (2015): Orden ECD/65/2015, de 21 de enero, por la que se describen las relaciones entre las competencias, los contenidos y los criterios de evaluación de la Educación Primaria, la Educación Secundaria Obligatoria y el Bachillerato, 29 enero 2015, nº 25, sec. I, pp. 6.986-7.003.
- [19] M. Bower (2006): “A taxonomy of task types in computing”, *Proceedings of the 11th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education (ITiCSE 2006)*, pp. 281-285.
- [20] K. Brennan y M. Resnick (2012): “Using artifact-based interviews to study the development of computational thinking in interactive media design”, *Annual American Educational Research Association Meeting*, Vancouver, BC, Canadá.
- [21] S. Carretero, R. Vuorikari y Y. Punie (2017): *DIGCOMP 2.1: The digital competence framework for citizens*, informe EUR 28558, Institute for Prospective Technological Studies, Joint Research Centre, Comisión Europea.
- [22] Committee for the Workshops on Computational Thinking (2010): *Report of a Workshop on the Scope and Nature of Computational Thinking*, National Academy of Sciences.
- [23] The Committee on European Computing Education (CECE), Informatics Europe & ACM Europe (2017): *Informatics Education in Europe: Are We All in the same Boat?*, <http://www.informatics-europe.org/component/phocadownload/category/10-reports.html?download=60:cece-report>.
- [24] Computer Science Teachers Association (CSTA) e International Society for Technology in Education (ISTE) (2011): *Operational Definition of Computational Thinking for K–12 Education*, <http://www.iste.org/docs/ct-documents/computational-thinking-operational-definition-flyer.pdf?sfvrsn=2>
- [25] I. Corradini, M. Lodi y E. Nardelli (2017): “Conceptions and misconceptions about computational thinking among Italian primary school teachers”, *Proceedings of the Fourteenth Annual ACM International Computing Education Research Conference (ICER 2017)*, pp. 136-144, DOI [10.1145/3105726.3106194](https://doi.org/10.1145/3105726.3106194).
- [26] P. J. Denning (2017): “Remaining trouble spots with computational thinking”, *Communications of the ACM*, 60(6):33-39, DOI [10.1145/2998438](https://doi.org/10.1145/2998438).

- [27] P. J. Denning, D. E. Comer, D. Gries, M. C. Mulder, A. Tucker, A. J. Turner y P. R. Young (1989): “Computing as a discipline”, *Communications of the ACM*, 32(1):9-23, enero 1989.
- [28] P. Denning y C. Martell (2015): *Great Principles of Computing*, MIT Press.
- [29] R. Diego Obregón (sin fecha), *Currículum «Competencia Digital» - Educación Primaria*, Los Salesianos, Santander.
- [30] C. Duncan, T. Bell y J. Atlas (2017): “What do the teachers think? Introducing computational thinking in the primary school curriculum”, *Proceedings of the 19th Australasian Conference on Computing Education (ACE 2017)*, pp. 65-74, DOI [10.1145/3013499.3013506](https://doi.org/10.1145/3013499.3013506).
- [31] A. Ferrari (2013): *DIGCOMP: A framework for developing and understanding digital competence in Europe*, informe EUR 26035, Institute for Prospective Technological Studies, Joint Research Centre, Comisión Europea.
- [32] Google, FECYT y Everis (2016): *Educación en ciencias de la computación en España 2015*, <https://www.fecyt.es/es/publicacion/educacion-de-las-ciencias-de-la-computacion-en-espana>.
- [33] T. Götschi, I. Sanders y V. Galpin (2003): “Mental models of recursion”, *Proceedings of the 34th SIGCSE Technical Symposium on Computer Science Education (SIGCSE 2003)*, pp. 346-350, DOI [10.1145/611892.612004](https://doi.org/10.1145/611892.612004).
- [34] M. Guzdial (2016): “Bringing computer science to U.S. schools, state by state”, *Communications of the ACM*, 59(5):24-25, mayo 2016, DOI [10.1145/2898963](https://doi.org/10.1145/2898963).
- [35] B. Hanks, S. Fitzgerald, R. McCauley, L. Murphy y C. Zander (2011): “Pair programming in education: a literature review”, *Computer Science Education*, 21(2): 135-173, DOI [10.1080/08993408.2011.579808](https://doi.org/10.1080/08993408.2011.579808).
- [36] O. Hazzan, T. Lapidot y N. Ragonis (2011): *Guide to Teaching Computer Science*, Springer.
- [37] D. Hemmendinger (2010): “A plea for modesty”, *ACM Inroads*, 1(2):4-7, junio 2010, DOI [10.1145/1805724.1805725](https://doi.org/10.1145/1805724.1805725).
- [38] Instituto Nacional de Tecnologías Educativas y Formación del Profesorado, INTEF (2017): *Marco Común de Competencia Digital Docente – Octubre 2017*.
- [39] Instituto Nacional de Tecnologías Educativas y Formación del Profesorado, INTEF (2018): *Situación de la competencia digital del alumnado en España – Comparativa de iniciativas*.
- [40] Instituto Nacional de Tecnologías Educativas y Formación del Profesorado, INTEF (2018): *Programación, robótica y pensamiento computacional en el aula: Situación en España – enero 2018*, <http://code.educalab.es/wp-content/uploads/2017/09/Pensamiento-Computacional-Fase-1-Informe-sobre-la-situación-en-España.pdf>.
- [41] Joint Informatics Europe & ACM Europe Working Group on Informatics Education (2013): *Informatics education: Europe cannot afford to miss the boat*, <http://europe.acm.org/iereport/ACMandIereport.pdf>.
- [42] P. Kemp (2014): *Computing in the National Curriculum – A guide for secondary teachers*, Computing at School.

- [43] R. Lister (2000): “On blooming first year programming, and its blooming assessment”, *Proceedings of the Australasian Conference on Computing Education (ACSE 2000)*, pp. 158-162, DOI [10.1145/359369.359393](https://doi.org/10.1145/359369.359393).
- [44] R. Lister, E. S. Adams, S. Fitzgerald, W. Fone, J. Hamer, M. Lindholm, R. McCartney, J. E. Moström, K. Sanders, O. Seppälä, B. Simon y L. Thomas (2004): ”A multi-national study of reading and tracing skills in novice programmers”, *ACM SIGCSE Bulletin*, 36(4):119-150, Dic. 2004.
- [45] M. López, J. Whalley, P. Robbins y R. Lister (2008): “Relationships between reading, tracing and writing skills in introductory programming”, *Proceedings of the Fourth international Workshop on Computing Education Research (ICER 2008)*, pp. 101-111, DOI [10.1145/1404520.1404531](https://doi.org/10.1145/1404520.1404531).
- [46] A. Luxton-Reilly (2016): “Learning to program is easy”, *Proceedings of the 21th ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE 2016)*, pp. 284-289, DOI [10.1145/2899415.2899432](https://doi.org/10.1145/2899415.2899432).
- [47] S. Y. Lye y J. H. L. Koh (2014): “Review on teaching and learning of computational thinking through programming: What is next for K-12?”, *Computers in Human Behavior*, 41:51-61, DOI [10.1016/j.chb.2014.09.012](https://doi.org/10.1016/j.chb.2014.09.012).
- [48] M. McCracken, Y.B.-D. Kolikant, V. Almstrum, C. Laxer, D. Diaz, L. Thomas, M. Guzdial, I. Utting, D. Hagan y T. Wilusz (2001): “A multi-national, multi-institutional study of assessment of programming skills of first-year CS students”, *ACM SIGCSE Bulletin*, 33(4):125-140, Dic. 2001.
- [49] J. Moreno León y G. Robles (2016): “Code to learn with Scratch? A systematic literature review”, *EDUCON 2016*, DOI [10.1109/EDUCON.2016.7474546](https://doi.org/10.1109/EDUCON.2016.7474546).
- [50] Official Journal of the European Union (2006): *Recommendation of the European Parliament and of the Council of 18 December 2006 on key competences for lifelong learning*, 2006/962/EC, 30 Dic. 2006, 394/10-18.
- [51] S. Papert (1980): *Mindstorms: Children, Computers, and Powerful Ideas*, Basic Books, New York.
- [52] D. Parsons y P. Haden (2006): “Parsons’ programming puzzles: A fun and effective learning tool for first programming courses”, *Proceedings of the 8th Australasian Conference on Computing Education (ACE 2006)*, pp. 157-163.
- [53] M. Resnick, J. Maloney, A. Monroy-Hernández, N. Rusk, E. Eastmond, K. Brennan, A. Millner, E. Rosenbaum, J. Silver, B. Silverman e Y Kafai (2009): “Scratch: Programming for all”, *Communications of the ACM*, 52(11):60-67, DOI [10.1145/1592761.1592779](https://doi.org/10.1145/1592761.1592779).
- [54] K. Sanders, M. Ahmadzadeh, T. Clear, S. H. Edwards, M. Goldweber, C. Johnson, R. Lister, R. McCartney, E. Patitsas y J. Spacco (2013): “The Canterbury QuestionBank: Building a repository of multiple-choice CS1 and CS2 questions”, *ITiCSE’13 Working Group Reports*, DOI [10.1145/2543882.2543885](https://doi.org/10.1145/2543882.2543885).
- [55] A. Schwill (1994): “Fundamental ideas of computer science”, *EATCS Bulletin*, 53:274-295.

- [56] A. Tucker (ed.) (1991): “Computing Curricula 1991”, *Communications of the ACM*, vol. 34, n°. 6, junio 1991.
- [57] S. Vogel, R. Santo y D. Ching, “Visions of computer science education: Unpacking arguments for and projected impacts of CS4All initiatives”, *Proceedings of the 2017 SIGCSE Technical Symposium on Computer Science Education (SIGCSE 2017)*, pp. 609-614, DOI [10.1145/3017680.3017755](https://doi.org/10.1145/3017680.3017755).
- [58] R. Vuorikari, Y. Punie, S. Carretero y L. Van den Brande (2016): *DIGCOMP 2.0: The digital competence framework for citizens*, informe EUR 27948, Institute for Prospective Technological Studies, Joint Research Centre, Comisión Europea.
- [59] J. L. Whalley, R. Lister, E. Thompson, T. Clear, P. Robbins, P. K. A. Kumar y C. Prasad (2006): “An Australasian study of reading and comprehension skills in novice programmers, using the Bloom and SOLO taxonomies”, *Proceedings of the Eighth Australasian Computing Education Conference (ACE 2006)*, pp. 243-252.
- [60] J. Wing (2006). “Computational thinking”, *Communications of the ACM*, 49(3):33-35, DOI [10.1145/1118178.1118215](https://doi.org/10.1145/1118178.1118215).
- [61] J. Wing (2010): “Computational thinking—What and why?”, *The Link Magazine*, Carnegie Mellon University, Pittsburgh, primavera 2010, <http://link.cs.cmu.edu/article.php?a=600>.