

PROGRAMACIÓN INFORMÁTICA

La tecnología impregna cualquier ámbito de la sociedad actual. Tanto en los puestos de trabajo como en los hogares y las escuelas, proliferan sistemas electrónicos que controlan muchos de los objetos que utilizamos de manera cotidiana. Esta cotidianeidad ha contribuido a mejorar la capacidad para interactuar con este tipo de productos tecnológicos. No obstante, y con carácter general, esta familiarización tiene lugar sin conocer los procesos que rigen su funcionamiento, los cuales en mayor o menor medida requieren del uso de la programación.

El aprendizaje de la programación representa un nuevo medio de alfabetización, que facilita la comprensión del mundo que nos rodea y permite intervenir en él y adoptar un papel activo. El alumnado deberá enfrentarse en el futuro con trabajos que ni siquiera existen en la actualidad; por ello, la introducción de la enseñanza de la programación informática le ofrece una ayuda para encarar este reto.

Si bien existen experiencias previas tales como la introducción en los años ochenta del lenguaje de programación *Logo*, principalmente en Estados Unidos, es en la actualidad cuando el desarrollo de la programación informática ha experimentado un impulso notable, como herramienta para el desarrollo de aplicaciones tecnológicas y con motivo de los probados beneficios cognitivos globales que promueve el hecho de aprender a programar.

La programación informática conlleva, por un lado, una forma determinada de pensar, analizar, organizar y relacionar ideas con el fin de resolver retos y, por otro, la identificación y resolución de errores que se puedan encontrar en dicho proceso. Todo ello requiere de un elevado grado de comprensión del problema de partida, de manera que seamos capaces de plasmar nuestra idea en instrucciones que comprenda el sistema informático.

El aprendizaje de la programación informática aporta importantes beneficios al desarrollo del razonamiento lógico, el pensamiento abstracto y algorítmico y la resolución de problemas complejos, lo cual a su vez prepara para solventar otras problemáticas presentes en la vida real. Todos los procesos anteriores, junto con el de tener que prestar atención a aspectos tales como las normas del lenguaje de programación, el patrón que debe seguir el programa, etc. potencia el grado de concentración.

Por otra parte, que dentro del campo de la programación informática un mismo problema pueda tener diferentes soluciones, tiene un efecto positivo a la hora de incentivar la experimentación y fomentar la autonomía, creatividad y motivación del alumnado.

Todas las destrezas señaladas en los párrafos anteriores están relacionadas con el futuro laboral de los alumnos; independientemente de la disciplina profesional que ocupen deberán enfrentarse a un entorno de trabajo en continuo cambio, que les exigirá la generación de soluciones originales que resuelvan situaciones imprevistas. En la actualidad, el escenario económico muestra una fuerte demanda de programadores en los próximos años, lo cual vincula el aprendizaje de la programación informática con una clara oportunidad en relación con el empleo. Además, la presencia de los contenidos de programación en el ámbito educativo desde edades tempranas es importante con el fin de evitar estereotipos de género en relación con las carreras STEM (*Science, Technology, Engineering and Mathematics*).

La materia Programación Informática se ha organizado en tres bloques. El primero de ellos, «Introducción a la programación», pretende introducir al alumnado en las bases del pensamiento computacional, mostrándole las diferentes estrategias que dicho paradigma emplea para la resolución de problemas complejos. Por otra parte, se trata de otorgar una introducción global a qué es un lenguaje de programación, qué tipos hay y qué sintaxis se utiliza en cada uno de ellos, tratando de hacer ver la lógica que se halla detrás.

El segundo bloque, «Entornos de programación gráfica mediante bloques» proporciona un primer contacto con la programación informática mediante herramientas sencillas que, a su vez, contienen elementos básicos de los lenguajes de alto nivel (variables, sentencias, operadores...). El alumnado aprenderá a realizar aplicaciones *apps* para dispositivos móviles (tabletas o *smartphones*), animaciones y videojuegos, utilizando entornos gráficos en donde cada acción viene determinada por bloques en paletas que conforman determinadas instrucciones de movimiento, acciones condicionales, sensorica y otras acciones que determinan, en su conjunto, herramientas de programación de una inestimable validez.

El último bloque, «Lenguajes de programación mediante código» constituye un primer paso a la hora de proveer al alumnado de las habilidades necesarias para afrontar el aprendizaje de lenguajes de programación por código, cuya importancia radica en la gran cantidad de aplicaciones y entornos audiovisuales en los que son empleados. El alumnado acometerá el aprendizaje de lenguajes

procedimentales que requieren la descripción exacta mediante códigos concretos, de todo tipo de acciones a realizar.

El aprendizaje de esta materia viene dado por una óptica activa y flexible, capaz de adaptarse a lenguajes de programación actuales y venideros, que capacite al alumnado para enfrentarse a las herramientas que se desarrollen en cada momento.

Debe impulsar un aprendizaje práctico basado en la superación de retos, favoreciendo así el desarrollo de habilidades de investigación, reflexión, toma de decisiones, creatividad, emprendimiento, aumento de la motivación y autonomía, ya que, a partir de un enunciado, y tras haber diseñado soluciones para acciones de menor complejidad, deben comprender el nuevo reto planteado y buscar una solución adecuada, consiguiendo así tomar la responsabilidad de aprender y comprobar el interés y la utilidad de lo que va diseñando.

Ha de fomentar el aprendizaje cooperativo. Tradicionalmente se ha asociado el aprendizaje de la programación informática a un modelo individualizado en el que el estudiante sólo interacciona con el ordenador y el reto planteado. Sin embargo, aunque la superación del reto puede tener una amplia componente individual, no está exento de la posibilidad de que el conocimiento de otros miembros coadyuve a la construcción del conocimiento propio, siguiendo la filosofía del aprendizaje cooperativo.

A su vez, este proceso de aprendizaje es idóneo para desarrollar las habilidades de trabajo en equipo, y entronca con la filosofía de conocimiento compartido que impregna el desarrollo de muchas aplicaciones informáticas *open source* o código abierto que han experimentado un vertiginoso avance en los últimos años, y que suponen un modelo de avance tecnológico en todo el mundo.

Es necesario, además, innovar, aplicando nuevos métodos de enseñanza como la metodología de clase invertida o *flipped classroom*. Este modelo traspasa aprendizajes fuera del aula, evitando las clases magistrales dentro de ella, y utiliza el tiempo de clase para desarrollar, dentro del aula y en compañía del profesor, otros procesos de aprendizaje que habitualmente los alumnos realizan en sus casas. Con ello se favorece la interacción profesor-alumno y hace posible un mejor control del proceso de enseñanza aprendizaje.

CUARTO CURSO

Contenidos	Criterios de evaluación	Estándares de aprendizaje evaluables
Bloque 1. Introducción a la programación		
<ul style="list-style-type: none"> - Pensamiento Computacional: Definición. Estrategias para la resolución de problemas mediante pensamiento computacional. - Lenguaje de programación: Definición. Lenguajes de alto y bajo nivel. Características. 	<ol style="list-style-type: none"> 1. Hacer uso del pensamiento computacional para formular problemas y sus soluciones. 2. Analizar los diferentes niveles de los lenguajes de programación. 	<ol style="list-style-type: none"> 1.1. Formula problemas, dividiéndolos en partes o pasos, haciendo uso de la informática para resolverlos. 1.2. Organiza de forma lógica la información y la representa a través de abstracciones. 1.3. Automatiza las soluciones haciendo uso del pensamiento computacional algorítmico y estableciendo pasos ordenados para llegar a la solución óptima. 1.4. Generaliza y transfiere el proceso de resolución de problemas para cualquier dominio. 2.1. Identifica las características de los lenguajes de programación de bajo y alto nivel. 2.2. Reconoce las diferencias entre las formas de ejecución de los programas informáticos
Bloque 2. Entornos de programación gráfica por bloques		
<ul style="list-style-type: none"> - Aplicaciones básicas de programación por bloques. Bloques de programación. Bucles. Realimentaciones. 	<ol style="list-style-type: none"> 1. Utilizar con destreza un entorno de programación gráfica por bloques, enfocado a la generación de juegos y animaciones. 	<ol style="list-style-type: none"> 1.1. Describe el proceso de desarrollo de una animación o un juego, y enumera las fases principales de su desarrollo.

Contenidos	Criterios de evaluación	Estándares de aprendizaje evaluables
<p>Variables de entorno y sensórica. Operadores. Bloques.</p> <p>- Programas de creación de aplicaciones móviles (<i>apps</i>) para dispositivos móviles (tabletas y <i>smartphones</i>). Paletas de bloques, eventos, variables, procedimientos y pantallas de interacción. Sensores. Simulación. Publicación. Licencias y uso de materiales en la Red Internet y propios.</p>	<p>2. Desarrollar, programar y publicar aplicaciones móviles sencillas en entornos de programación por bloques.</p> <p>3. Discrimina la licencia pertinente a la hora de reutilizar contenidos de Internet, así como a la hora de publicar materiales propios.</p>	<p>1.2. Emplea, con facilidad, las diferentes herramientas básicas del entorno de programación.</p> <p>1.3. Sitúa y mueve objetos en una dirección dada.</p> <p>1.4. Inicia y detiene la ejecución de un programa.</p> <p>1.5. Modifica, mediante la edición, la apariencia de objetos. Crea nuevos objetos: actores, fondos y sonidos.</p> <p>1.6. Maneja, con soltura, los principales grupos de bloques del entorno.</p> <p>1.7. Utiliza, con facilidad, los comandos de control de ejecución: condicionales y bucles.</p> <p>1.8. Emplea de manera adecuada variables y listas.</p> <p>1.9. Usa, con soltura, la interacción entre los elementos de un programa.</p> <p>1.10. Analiza el funcionamiento de un programa a partir de sus bloques.</p> <p>1.11. Identifica y considera las implicaciones del "diseño para todos" para los programas que realiza.</p> <p>1.12. Crea bloques que pueden ser llamados en múltiples ocasiones.</p> <p>2.1. Describe el proceso de diseño de una aplicación para dispositivos móviles y las fases principales de su desarrollo.</p> <p>2.2. Utiliza con precisión las diferentes herramientas del entorno de desarrollo.</p> <p>2.3. Distingue los diferentes tipos de datos y sus formas de presentación y almacenamiento.</p> <p>2.4. Clasifica los objetos disponibles, sus métodos y eventos.</p> <p>2.5. Identifica las posibilidades de interacción con los sensores de los que dispone un terminal móvil.</p> <p>2.6. Reconoce y evalúa las implicaciones del "diseño para todos" para las <i>apps</i> que realiza.</p> <p>2.7. Desarrolla aplicaciones informáticas para su ejecución en dispositivos móviles utilizando diferentes sensores y elementos de interfaz.</p> <p>2.8. Visualiza los resultados de su programación en simuladores asociados a las diferentes plataformas y sistemas operativos.</p> <p>2.9. Describe las características y normas de diferentes plataformas</p>

Contenidos	Criterios de evaluación	Estándares de aprendizaje evaluables
		para la publicación de aplicaciones móviles. 3.1. Elige la licencia más adecuada en relación al uso a que va a destinar materiales reutilizables en la Red Internet y a la publicación de materiales propios.
Bloque 3. Lenguajes de programación mediante código.		
Elementos de un lenguaje de programación. Sintaxis. Variables. Estructuras de control. Vectores. Funciones. Compiladores. Depuración de errores.	1. Desarrollar programas en un lenguaje de programación textual, como <i>Python</i> , <i>PHP</i> , <i>Processing</i> , <i>JavaScript</i> , etc.	1.1. Utiliza de manera adecuada los diferentes tipos de datos y estructuras. 1.2. Usa de forma adecuada estructuras de control de ejecución. 1.3. Analiza los diferentes problemas, descomponiéndolos en elementos más sencillos. 1.4. Elabora algoritmos mediante diagramas de flujo para resolver problemas concretos de programación. 1.5. Codifica un programa fuente en el lenguaje de programación aprendido y crea un programa ejecutable. 1.6. Emplea con facilidad el sistema de almacenamiento y archivos. 1.7. Analiza el funcionamiento de programas y algoritmos a partir del código. 1.8. Depura el programa: identifica los errores de programación mediante la realización de pruebas y corrige los errores de ejecución. 1.9. Documenta adecuadamente los algoritmos y programas desarrollados incorporando comentarios. 1.10. Adapta programas a partir de nuevos requerimientos establecidos en el diseño. 1.11. Utiliza librerías de funciones disponibles en Internet.