

PROGRAMACIÓN Y COMPUTACIÓN

La materia de Programación y Computación es una materia de libre configuración autonómica que se oferta en el segundo curso de Bachillerato. Las Ciencias de la Computación son la disciplina dedicada al estudio, diseño y construcción de programas y sistemas informáticos, sus principios y prácticas, aplicaciones y el impacto que éstas tienen en nuestra sociedad. Se trata de una materia con un cuerpo de conocimiento bien establecido, que incluye un marco de trabajo centrado en la resolución de problemas y en la creación de conocimiento. Por otro lado, la Programación, su elemento más representativo, es considerada por la Comisión Europea la competencia del siglo XXI, una nueva forma de alfabetización, fundamental para la comprensión de la Sociedad del Conocimiento.

Las Ciencias de la Computación no se circunscriben al ámbito informático, a día de hoy, tienen un enorme impacto en todas las disciplinas: ya sea biología, química, física, ingeniería, economía o geografía. A modo de ejemplo, en las ciencias de la salud, la computación permite que se investigue sobre una enorme cantidad de datos médicos de múltiples fuentes y que se puedan tomar decisiones correctas, en el momento adecuado, para salvar vidas.

Aunque el software es intangible, se trata de una de las creaciones más complejas de la humanidad, y las personas que profundicen en este conocimiento estarán mejor preparadas para integrarse activamente en un mundo en continuo proceso de transformación, en el cual la computación es motor de cambio.

La Programación y Computación, y, las Tecnologías de la Información y Comunicación son materias complementarias, mientras la primera enseña al alumnado a ser creador de aplicaciones informáticas, la segunda tiene como objetivo enseñar el uso productivo y creativo de las mismas. Hay que señalar, además, que aprender Ciencias de la Computación permite conceptualizar y comprender mejor los sistemas informáticos, y por tanto hacer un uso más productivo de ellos.

El valor educativo de la materia de Programación y Computación es doble: por un lado permite que los alumnos y las alumnas sean capaces de idear, planificar, diseñar y crear software como una herramienta que permite cambiar el mundo, y por otro, desarrollar una serie de capacidades cognitivas integradas en el denominado pensamiento computacional. Esta forma de pensar enseña a razonar sobre sistemas y problemas mediante un conjunto de técnicas y prácticas bien definidas que permiten su análisis, modelado y resolución. El pensamiento computacional engloba una serie de principios, que definen el marco de trabajo, intrínsecamente competencial, de la disciplina y que son la creatividad, la abstracción, el análisis de problemas, el pensamiento lógico y crítico, la comunicación y la colaboración.

Considerando ambos planteamientos, se establece que el eje vertebrador de la materia de Programación y Computación sea el “desarrollo en equipo de aplicaciones informáticas mediante el uso del pensamiento computacional”. El objetivo es que el proceso de enseñanza-aprendizaje sea práctico, aplicado y contextualizado, se integren las competencias clave y permita desarrollar las capacidades del alumnado mediante una metodología activa y participativa.

El cuerpo de conocimiento de las Ciencias de la Computación es extenso y por tanto se pretende que la materia de Programación y Computación en segundo de Bachillerato sea introductory. Debe servir para responder a una serie de cuestiones articuladas en torno al pensamiento computacional y a temáticas centrales en la disciplina como son los algoritmos, la programación, los datos y la información, Internet y el impacto de la computación en nuestra sociedad. Entre otras, se pueden destacar:

¿Cómo influye el proceso creativo en el desarrollo de sistemas informáticos? ¿Cómo se puede emplear la computación para crear conocimiento? ¿Cómo se implementan y ejecutan los algoritmos en los sistemas informáticos? ¿Cómo se desarrollan aplicaciones para ayudar a personas, organizaciones o la sociedad en su conjunto y resolver problemas? ¿Cómo posibilita la abstracción el desarrollo de programas? ¿Qué consideraciones y compromisos existen en la manipulación de datos? ¿Qué aspectos del diseño de Internet han permitido construir un medio de comunicación abierto y global? ¿Qué consecuencias tiene la ciberseguridad para los usuarios de Internet? ¿Cómo es fomentada la innovación por la computación? ¿Qué efectos beneficiosos y perjudiciales tiene la computación para la sociedad?

Por otro lado, la Programación y la Computación tiene un ámbito de aplicación multidisciplinar, integra

conocimientos de otras materias como Matemáticas, Física, etc. y permite trabajar conocimientos relativos al patrimonio de Andalucía o a los elementos transversales del currículo como objetos de las aplicaciones informáticas a desarrollar.

Además de ello, desde la materia de Programación y Computación se debe, prioritariamente, impulsar la igualdad real y efectiva de sexos, corrigiendo estereotipos que provocan que la materia sea poco popular entre las mujeres; emplear modelos de utilidad social y sostenibilidad en el desarrollo de las aplicaciones; fomentar una utilización crítica, responsable, segura y autocontrolada en el uso de las tecnologías informáticas y de comunicaciones; promover un clima de respeto, convivencia y tolerancia en los medios de comunicación electrónicos, prestando especial atención a cualquier forma de acoso, rechazo o violencia; incentivar la utilización de herramientas de software libre y minimizar el riesgo de brecha digital.

La materia de Programación y Computación contribuye al desarrollo de las competencias clave. De forma general, se considera que la competencia en comunicación lingüística (CCL) se fomenta mediante la interactuación con otros interlocutores y a través de textos en múltiples modalidades, formatos y soportes; la competencia matemática y las competencias básicas en ciencia y tecnología (CMCT) empleando el razonamiento matemático y sus herramientas, aplicando métodos propios de la racionalidad científica y destrezas tecnológicas; la competencia digital (CD) usando de forma creativa, crítica y segura las tecnologías de la información y comunicación; la competencia de aprender a aprender (CAA) desarrollando la habilidad para iniciar, organizar y persistir en el aprendizaje; las competencias sociales y cívicas (CSC) desarrollando la capacidad para interpretar fenómenos y problemas sociales y colaborando con sus compañeros; el sentido de la iniciativa y espíritu emprendedor desarrollando la habilidad para transformar ideas y reconociendo oportunidades existentes para las actividades personales y profesionales; la competencia en conciencia y expresiones culturales (CEC) desarrollando la capacidad estética y creadora, para poder utilizarlas como medio de comunicación y expresión personal.

Objetivos

1. Comprender el impacto que la computación tiene en la sociedad actual, sus aspectos positivos y negativos, y su influencia en la innovación, la comunicación y el conocimiento.
2. Producir programas informáticos plenamente funcionales utilizando las principales estructuras de un lenguaje de programación, describiendo cómo los programas implementan algoritmos y evaluando su corrección.
3. Integrarse en un equipo de desarrollo de software que sea capaz de afrontar proyectos de poca envergadura, colaborando y comunicándose con sus compañeros, fomentando sus habilidades sociales mediante la búsqueda del consenso, la negociación y la resolución de conflictos.
4. Desarrollar la capacidad de abstracción usando modelos para describir fenómenos, conociendo diferentes tipos de representaciones de datos y escribiendo programas que generalicen funcionalidades.
5. Emplear la creatividad en el desarrollo de aplicaciones informáticas para resolver un problema, o como forma de expresión personal, analizando su usabilidad, funcionalidad e idoneidad al contexto.
6. Recopilar, almacenar y procesar datos con el objetivo de encontrar patrones, descubrir conexiones y resolver problemas, utilizando herramientas de análisis y visualización que permitan extraer información, presentarla y construir conocimiento.
7. Analizar cómo la representación, el almacenamiento, la seguridad y la transmisión de datos requiere de manipulación computacional, y comprender los riesgos de seguridad y privacidad existentes cuando se trata de información personal.
8. Comprender el funcionamiento y las características de Internet, analizando los principios de diseño y los mecanismos de abstracción que han permitido su evolución y crecimiento, identificando aspectos relativos a ciberseguridad y sus posibles soluciones.
9. Comprender los principios del desarrollo web, creando aplicaciones web sencillas con acceso a una base de datos, utilizando tecnologías del servidor y aplicando mecanismos para separar la presentación de la lógica.

10 Explorar la computación física, construyendo un sistema hardware y software que interactúe con el medio físico, detectando y respondiendo a cambios en el mundo real, comprendiendo las diferencias entre los mundos digital y analógico.

Estrategias metodológicas

Las Ciencias de la Computación son una disciplina dedicada al estudio, diseño y construcción de aplicaciones y sistemas informáticos y por tanto su metodología debe centrarse en abordar los principios fundamentales y técnicas sobre los que se crean estos sistemas, abandonando la perspectiva de usuario.

La creatividad, el pensamiento lógico y crítico, la capacidad de resolución de problemas y la abstracción son habilidades cognitivas esenciales que forman parte del denominado pensamiento computacional y que deberán ser desarrolladas y refinadas de manera progresiva durante el curso, empleando mecanismos tales como el modelado, la descomposición de problemas, la generalización o el reconocimiento de patrones. La programación ofrece una forma concreta y tangible de materializar la idea de abstracción.

Además de la competencia digital, desarrollar aplicaciones debe promover que los alumnos y las alumnas sean capaces de expresarse correctamente de forma oral, presentando en público sus creaciones y propuestas y comunicándose con sus compañeros y sus compañeras de manera respetuosa y cordial, de redactar la documentación asociada al desarrollo y de consolidar el hábito de la lectura; aplicar conocimientos matemáticos, científicos y tecnológicos en el diseño, implementación y prueba de las aplicaciones; aprender a aprender ante problemas complejos, con los que no están familiarizados, que les obliguen a movilizar sus destrezas personales y sociales, en un ámbito de conocimiento en continuo proceso de cambio; trabajar individualmente y en equipo de manera autónoma, construyendo y compartiendo el conocimiento, llegando a acuerdos sobre las responsabilidades propias y las de sus compañeros; tomar decisiones, planificar, organizar el trabajo y evaluar los resultados; y crear y entender las posibilidades que el software ofrece como herramienta de expresión personal y cultural, y usarlo de forma segura y responsable.

Durante el curso, el alumnado deberá realizar proyectos cooperativos de desarrollo de software, encuadrados en los bloques de contenidos de la materia. Estos proyectos abarcarán las etapas de análisis, diseño, implementación y verificación del ciclo de vida del software. En ellos, se podrían emplear métodos y técnicas de desarrollo “ágiles”, basadas en iteraciones incrementales, en las que se van añadiendo nuevas funcionalidades al software en cada iteración. En estos proyectos el alumnado deberá idear, crear y presentar una aplicación informática de interés común a todos los miembros de su equipo. Asimismo, cada alumno y cada alumna será responsable de desarrollar una parte de la aplicación dentro de su equipo, hacer un seguimiento del desarrollo de las otras partes y de trabajar en la integración de los diferentes componentes en el producto final. Igualmente, cada equipo deberá almacenar las diferentes versiones del programa, redactar y mantener la documentación asociada al proyecto (análisis, diseño, codificación y verificación), elaborar un breve video sobre su funcionamiento y presentarlo a sus compañeros. De manera individual cada miembro del grupo, deberá redactar un diario sobre el desarrollo del proyecto y contestar a dos cuestionarios finales: uno sobre su trabajo individual y otro sobre el trabajo en equipo.

Por otro lado, un programa puede ayudarnos a resolver un problema, a promover una innovación o a expresar un interés personal. Por ello, los alumnos y las alumnas deberían desarrollar software en base a sus propias motivaciones, disponiendo de la oportunidad de materializar sus ideas y de cambiar el mundo en el que viven. Un enfoque multidisciplinar, que incluya temáticas de otras materias y los elementos transversales del currículo constituyen un punto de partida para este planteamiento. Entre otros, el alumnado podría desarrollar aplicaciones relacionadas con los derechos y libertades fundamentales; la convivencia y el respeto; la prevención del acoso escolar o de la discriminación contra personas con discapacidad; la igualdad efectiva entre mujeres y hombres; la convivencia intercultural; los hábitos de vida saludable; la educación para el consumo; la utilización crítica y racional de las tecnologías de información y comunicación y de los medios audiovisuales, la convivencia vial, etc.

Por último, los entornos de aprendizaje online dinamizan el proceso de enseñanza-aprendizaje, facilitando tres aspectos clave: la interacción con el alumnado, la atención personalizada y la evaluación. Con el objetivo de

orientar el proceso educativo, ajustarse al nivel competencial inicial del alumnado y a respetar los distintos ritmos de aprendizaje, se propone la utilización de entornos de aprendizaje online. Estos entornos deberían incluir formularios automatizados que permitan la autoevaluación y coevaluación del aprendizaje por parte de los alumnos y alumnas, la evaluación del nivel inicial, de la realización de los proyectos, del desarrollo competencial y del grado de cumplimiento de los criterios. Así como, repositorios de aplicaciones, documentación y tareas, que permitan hacer un seguimiento del trabajo individual y grupal de los estudiantes a lo largo del curso y visualizar su evolución. Además, se recomienda usar herramientas para la gestión de proyectos, software de productividad colaborativo y de comunicación, y otras aplicaciones propias de la disciplina como entornos de desarrollo integrados y software para el control de versiones.

Contenidos y criterios de evaluación

Programación y Computación. 2º Bachillerato

Bloque 1. Representación digital de la información.

Sociedad del Conocimiento. Papel de la Computación en la innovación tecnológica actual. Impacto social y económico de la Computación en nuestro mundo. Representación binaria de la información: el bit, el byte. Almacenamiento, transmisión y tratamiento básico de la información en binario: números, texto, imágenes, ficheros. Representación hexadecimal.

Criterios de evaluación

1. Describir el impacto de la computación en la sociedad y los aspectos positivos y negativos del mismo. CD, CSC, CED.
2. Explicar cómo la computación afecta a la innovación en otras disciplinas y posibilita la comunicación, la interacción y el conocimiento. CCL, CD, SIEP.
3. Describir la variedad de mecanismos de abstracción empleados para representar datos. CMCT, CD, CAA.
4. Explicar cómo se representan los datos digitalmente en forma de secuencias binarias. CD, CMCT, CCL.

Bloque 2. Programación.

Lenguajes de programación: Estructura de un programa informático y elementos básicos del lenguaje. Tipos de lenguajes. Tipos básicos de datos. Constantes y variables. Operadores y expresiones. Comentarios. Estructuras de control. Condicionales e iterativas. Profundizando en un lenguaje de programación: Estructuras de datos. Funciones y bibliotecas de funciones. Reutilización de código. Facilidades para la entrada y salida de datos de usuario. Manipulación de archivos. Orientación a objetos: Clases, objetos y constructores. Herencia. Subclases y superclases. Polimorfismo y sobrecarga. Encapsulamiento y ocultación. Bibliotecas de clases. Metodologías de desarrollo de software: Enfoque Top-Down, fragmentación de problemas y algoritmos. Pseudocódigo y diagramas de flujo. Depuración. Entornos de desarrollo integrado. Ciclo de vida del software. Análisis, Diseño, Programación y Pruebas. Trabajo en equipo y mejora continua. Control de versiones.

Criterios de evaluación

1. Descomponer problemas complejos en otros más simples, e idear modelos abstractos de los mismos y algoritmos que permiten implementar una solución computacional. CMCT, CD.
2. Identificar, elegir y operar adecuadamente los diferentes tipos de datos en el programa. CMCT, CD.
3. Escribir programas, convenientemente estructurados y comentados, que recogen y procesan la información procedente de diferentes fuentes y generan la correspondiente salida. CMCT, CD, CCL.
4. Escribir programas que instancian y usan objetos de clases propias y ajenas, y utilizan bibliotecas de funciones u objetos. CMCT, CD.
5. Identificar y aplicar los principales pasos del ciclo de vida de una aplicación, trabajando de forma colaborativa en equipos de desarrollo. CMCT, CD, SIEP, CSC.
6. Aplicar la creatividad al proceso de desarrollo de software, transformando ideas en aplicaciones. CD, CED,

CSC.

7. Elegir y utilizar IDE, depuradores y herramientas de control de versiones de código. CMCT, CD, SIEP.
8. Diseñar y probar programas propios o ajenos, elaborando la correspondiente documentación. CMCT, CD, SIEP, CCL.

Bloque 3. Datos e Información.

Almacenamiento de la información: Ficheros. Bases de datos relacionales. Sistemas gestores de bases de datos. Diseño conceptual. Diagramas entidad-relación. Normalización hasta 3FN. Definición y manipulación. Comandos básicos de SQL: create, insert, delete, select, update. Big data: Volumen y variedad de datos. Datos estructurados, no estructurados y semiestructurados. Introducción a las bases de datos NoSQL. Recogida y almacenamiento. Seguridad y privacidad. Extracción y limpieza. Análisis y visualización.

Criterios de evaluación

1. Describir los sistemas lógicos de almacenamiento y sus características básicas. CD, CCL.
2. Diseñar, crear y manipular una base de datos relacional sencilla, utilizando comandos básicos de SQL. CD, CMCT, CAA.
3. Conocer las posibilidades de las bases de datos para el manejo de grandes cantidades de información. CMCT, CSC, CD.
4. Recoger, almacenar y procesar datos para encontrar patrones, descubrir conexiones, y resolver problemas. CMCT, CD, CAA, SIEP.
5. Emplear herramientas de análisis y visualización para obtener información y conocimiento. CD, CAA, CSC, SIEP.
6. Describir los aspectos relacionados con la seguridad y privacidad en la gestión de datos. CD, CSC, CMCT.

Bloque 4. Internet.

Diseño: Organización y estructura. Modelo TCP/IP. Direccionamiento IP. Funcionamiento: Enrutamiento. Modelo cliente/servidor. Protocolo de Control de las Transmisiones (TCP). Sistema de Nombres de Dominio (DNS). Protocolo de Transferencia de Hipertexto (HTTP). Seguridad: Ciberseguridad. Criptografía. Cifrado de clave pública. Ciberseguridad en el mundo real, Hacking. Desarrollo web: Lenguaje de marcas de hipertexto (HTML), estructura, etiquetas y atributos, formularios, multimedia y gráficos. Hoja de estilo en cascada (CSS), diseño adaptativo y plantillas. Herramientas de diseño web. Visión general de los lenguajes de scripts. Introducción a la programación en entorno servidor. Acceso a bases de datos.

Criterios de evaluación

1. Explicar la estructura y características de Internet como una red de sistemas autónomos que facilita la comunicación global. CMCT, CD, CSC.
2. Identificar los componentes básicos de Internet y los mecanismos de abstracción que permiten su funcionamiento. CMCT, CD, CAA.
3. Explicar los principios de seguridad en Internet basados en la criptografía, el cifrado y las técnicas de autenticación, así como identificar amenazas y riesgos de seguridad. CMCT, CD, CSC.
4. Utilizar los lenguajes de marcado y presentación para la elaboración de páginas web. CMCT, CD, CCL.
5. Emplear herramientas de diseño web, utilizando plantillas, teniendo en cuenta aspectos relativos al diseño adaptativo. CD, SIEP, CED.
6. Diseñar, programar y probar una aplicación web sencilla con acceso a una base de datos, mediante un lenguaje de script en el entorno servidor. CMCT, CD, CAA, SIEP.

Bloque 5. Computación física. Robótica.

Programación de dispositivos inteligentes. Características principales de los robots: cuerpo, control y comportamiento. Microcontroladores, entrada/salida, sensores, actuadores, RFID. El Internet de las Cosas: Aplicaciones. De la casa inteligente a la ciudad Inteligente.

Criterios de evaluación

1. Identificar qué criterios determinan si un dispositivo es un robot o no. CSC, CD.
2. Describir los principios de funcionamiento de Internet de las Cosas. CMCT, CD, CAA.
3. Diseñar, programar y probar una aplicación que lea datos de un sensor, los procese, y como resultado, ejecute un actuador. CMCT, CD, CED.