

Etapa	Educación Secundaria Obligatoria
Curso	Segundo de Bachillerato
Asignatura	Programación y Computación

Descripción de la asignatura

Programación y Computación tiene una doble finalidad: por un lado, permite que el alumnado sea capaz 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 a la disciplina como son la creatividad, la abstracción, el análisis de problemas, el pensamiento lógico y crítico, la comunicación y la colaboración.

Las ciencias de la computación están dedicadas al estudio, diseño y construcción de programas y sistemas informáticos, sus principios, prácticas y aplicaciones. Se trata de un cuerpo de conocimiento bien establecido que incluye un marco de trabajo centrado en la resolución de problemas y en la creación de conocimiento.

Las ciencias de la computación no se circunscriben al ámbito informático, sino que, a día de hoy, tienen un enorme impacto en todas las disciplinas: Biología, Química, Física, Ingeniería, Economía o Geografía.

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.

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”, siempre desde el diseño del proceso de enseñanza-aprendizaje de manera práctica, aplicada y contextualizada, integrando las competencias clave y permitiendo desarrollar las capacidades del alumnado mediante una metodología activa y participativa.

El cuerpo de conocimiento de las ciencias de la computación 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, e Internet.

Competencias específicas

1. Desarrollar la capacidad de abstracción, producir programas informáticos funcionales e integrarse en un equipo de desarrollo de software que sea capaz de afrontar proyectos acordes al nivel de desarrollo del alumnado, fomentando sus habilidades sociales y aplicando la creatividad.
2. Recopilar y procesar datos que ayuden en la resolución de un problema, analizando cómo su almacenamiento, transmisión y presentación se benefician de la manipulación computacional.
3. Desarrollar aplicaciones web sencillas con acceso a una base de datos utilizando html, css y un lenguaje de script, elaborando páginas web con el fin de programar de manera accesible.
4. 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, para comprender las diferencias entre los mundos digital y analógico.

Procedimientos e instrumentos de evaluación

La normativa actual establece que, para poder superar una asignatura hay que medir el grado de consecución de las competencias específicas tomando como referentes los criterios de evaluación de cada una de las materias. Por tanto, en la asignatura de Programación y Computación de 2º de Bachillerato, para poder adquirir las competencias específicas que establece la normativa, se debe superar los criterios de evaluación asociados a las mismas.

Durante el desarrollo del curso se realizarán una serie de actividades en las que se utilizarán diferentes instrumentos para su evaluación, los cuales aportarán una calificación objetiva sobre la actividad desarrollada. Cada una de esas actividades estará asociada a un criterio de evaluación, y todas ponderarán lo mismo, por lo que para saber la nota que se obtiene al evaluar un criterio de evaluación, bastará con hacer la nota media de todas las actividades donde se evalúa el citado criterio.

Para superar la asignatura, habrá que tener al menos una calificación de 5 en cada uno de los criterios de evaluación, siendo la nota final de la asignatura la media aritmética de la calificación obtenida en todos los criterios de evaluación.

INFORMACIÓN SOBRE EL PLAGIO

1. El I.E.S. fomentará el respeto a la propiedad intelectual y transmitirá a los estudiantes que el plagio es una práctica contraria a los principios que rigen la educación secundaria.
 2. El plagio, entendido como la presentación de un trabajo u obra hecho por otra persona como propio o la copia de textos sin citar su procedencia y dándolos como de elaboración propia, conllevará automáticamente la calificación numérica de cero en el trabajo en el que se hubiera detectado. Esta consecuencia debe entenderse sin perjuicio de las responsabilidades disciplinarias en las que pudieran incurrir los estudiantes que plagien
-

Saberes básicos

A. Programación

PRYC.2.A.1. Lenguajes de programación.

PRYC.2.A.1.1. Tipos de lenguajes. Estructura de un programa informático y elementos básicos del lenguaje. Tipos básicos de datos. Constantes y variables. Operadores y expresiones. Comentarios.

PRYC.2.A.1.2. Estructuras de control condicionales e iterativas. Estructuras de datos.

PRYC.2.A.1.3. Funciones y reutilización de código. Manipulación de archivos.

PRYC.2.A.2. Orientación a objetos.

PRYC.2.A.2.1. Clases, objetos y constructores. Sobrecarga, encapsulamiento y ocultación.

PRYC.2.A.2.2. Herencia. Subclases y superclases. Interfaces. Polimorfismo.

PRYC.2.A.3. Ciclo de vida del software.

PRYC.2.A.3.1. Metodologías de desarrollo de software.

PRYC.2.A.3.2. Enfoque Top-Down, fragmentación de problemas y algoritmos.

PRYC.2.A.3.3. Pseudocódigo y diagramas de flujo.

PRYC.2.A.3.4. Desarrollo iterativo.

PRYC.2.A.3.5. Entornos de desarrollo integrado.

PRYC.2.A.3.6. Análisis, Diseño, Programación y Pruebas. Depuración.

PRYC.2.A.3.7. Control de versiones.

PRYC.2.A.3.8. Trabajo en equipo.

B. Datos e Información.

PRYC.2.B.1. Bases de datos relacionales.

PRYC.2.B.1.1. Sistemas gestores de bases de datos. Ventajas con respecto a los archivos.

PRYC.2.B.1.2. Diseño de bases de datos relacionales. Diagramas entidad-relación, esquema relacional y normalización.

PRYC.2.B.1.3. Creación y manipulación de bases de datos relacionales. Comandos básicos de SQL: create, insert, delete, select, update.

PRYC.2.B.2. Big data.

PRYC.2.B.2.1. Volumen y variedad de datos. Datos estructurados, no estructurados y semiestructurados.

PRYC.2.B.2.2. Introducción a las bases de datos NoSQL.

C. Desarrollo web.

PRYC.2.C.1. Lenguajes descriptivos.

PRYC.2.C.1.1. Lenguaje de marcas de hipertexto (HTML), documentos, etiquetas, estructura, elementos, y atributos.

PRYC.2.C.1.2. Títulos, texto, listas, tablas, formularios y multimedia.

PRYC.2.C.1.3. Hojas de estilo en cascada (CSS). Reglas de estilo. Selectores. Declaraciones. Propiedades y Valores.

PRYC.2.C.1.4. El modelo de cajas. Diseño adaptativo.

PRYC.2.C.2. Lenguajes de programación.

PRYC.2.C.2.1. Visión general de los lenguajes de scripts.

PRYC.2.C.2.2. Programación en entorno cliente.

PRYC.2.C.2.3. Introducción a la programación en entorno servidor.

PRYC.2.C.2.4. Acceso a bases de datos. Interfaz de programación de aplicaciones con servicios web (REST APIs).

D. Computación física y robótica.

PRYC.2.D.1. Robótica.

PRYC.2.D.1.1 Características principales de los robots: cuerpo, control y comportamiento.

PRYC.2.D.1.2. Microcontroladores, entrada/salida, sensores y actuadores.

PRYC.2.D.1.3. Programación de dispositivos inteligentes.

PRYC.2.D.2. El Internet de las Cosas.

PRYC.2.D.2.1. Aplicaciones. Smart Cities.

Competencias específicas, criterios de evaluación y saberes básicos

<p>1. Desarrollar la capacidad de abstracción, producir programas informáticos funcionales e integrarse en un equipo de desarrollo de software que sea capaz de afrontar proyectos acordes al nivel de desarrollo del alumnado, fomentando sus habilidades sociales y aplicando la creatividad.</p>	<p>1.1. Transformar ideas en aplicaciones de forma creativa, descomponiendo problemas complejos en otros más simples e ideando modelos abstractos de los mismos y algoritmos que permitan implementar una solución computacional.</p> <p>1.2. Escribir programas, convenientemente estructurados y comentados, que recogen y procesan la información procedente de diferentes fuentes y generan la correspondiente salida.</p> <p>1.3. Identificar y aplicar los principales pasos del ciclo de vida de una aplicación, y trabajar de forma colaborativa en equipos de desarrollo, utilizando IDE's, depuradores y herramientas de control de versiones de código</p>	<p>PRYC.2.A.3.1. PRYC.2.A.3.2. PRYC.2.A.3.3. PRYC.2.A.3.4. PRYC.2.A.1.1. PRYC.2.A.1.2. PRYC.2.A.1.3. PRYC.2.A.2.1. PRYC.2.A.2.2. PRYC.2.A.3.5. PRYC.2.A.3.6. PRYC.2.A.3.7. PRYC.2.A.3.8.</p>
<p>2. Recopilar y procesar datos que ayuden en la resolución de un problema, analizando cómo su almacenamiento, transmisión y presentación se benefician de la manipulación computacional.</p>	<p>2.1. Explotar las posibilidades de las bases de datos para la recogida y procesamiento de grandes cantidades de datos en la búsqueda de patrones y conexiones que faciliten la resolución de problemas computacionales.</p>	<p>PRYC.2.B.1.1. PRYC.2.B.1.2. PRYC.2.B.1.3. PRYC.2.B.2.1. PRYC.2.B.2.2.</p>
<p>3. Desarrollar aplicaciones web sencillas con acceso a una base de datos utilizando html, css y un lenguaje de script, elaborando páginas web con el fin de programar de manera accesible.</p>	<p>3.1. Utilizar los lenguajes de marcado y estilos para la creación de páginas web, teniendo en cuenta aspectos relativos al diseño adaptativo.</p> <p>3.2. Diseñar, programar y probar una aplicación web sencilla con acceso a una base de datos, utilizando un lenguaje de script.</p>	<p>PRYC.2.C.1.1. PRYC.2.C.1.2. PRYC.2.C.1.3. PRYC.2.C.1.4. PRYC.2.C.2.1. PRYC.2.C.2.2. PRYC.2.C.2.3. PRYC.2.C.2.4.</p>
<p>4. 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, para comprender las diferencias entre los mundos digital y analógico.</p>	<p>4.1. Diseñar, programar y probar una aplicación que lea datos de un sensor, los procese, y como resultado, ejecute un actuador.</p>	<p>PRYC.2.D.1.1. PRYC.2.D.1.2. PRYC.2.D.1.3. PRYC.2.D.2.1.</p>