

<b>PROGRAMACIÓN DIDÁCTICA.</b>	
<b>I.E.S. JULIO VERNE</b>	<b>DEPARTAMENTO DE INFORMÁTICA</b>
<b>BACHILLERATO</b>	<b>2º CURSO</b>
<b>ASIGNATURA: PROGRAMACIÓN Y COMPUTACIÓN</b>	

<b>CURSO:</b>	<b>2023/24</b>
<b>PROFESORES:</b>	<b>ANA M<sup>a</sup> GÓMEZ SIRÉS</b>

## ÍNDICE

<b>0.- Introducción.....</b>	<b>2</b>
<b>1.- Relación entre elementos curriculares.....</b>	<b>4</b>
<b>1.1.- Criterios de evaluación.....</b>	<b>4</b>
<b>1.2.- Saberes básicos.....</b>	<b>5</b>
<b>1.3.- Tabla 1.....</b>	<b>7</b>
<b>2.- Evaluación (Tabla 2).....</b>	<b>8</b>
<b>3.- Temporalización.....</b>	<b>10</b>
<b>4.- Situaciones de aprendizaje.....</b>	<b>11</b>
<b>5.- Medidas de atención a la diversidad.....</b>	<b>12</b>

## **0.- INTRODUCCIÓN.**

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.

La materia de Programación y Computación contribuye al desarrollo de las competencias clave a través de sus competencias específicas. De forma general, se considera que la competencia en comunicación lingüística (CCL) se fomenta mediante la interacción con otros interlocutores y a través de textos en múltiples modalidades, formatos y soportes; la competencia matemática y competencia en ciencia, tecnología e ingeniería (STEM), 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 personal, social y de aprender a aprender (CPSAA); la competencia emprendedora (CE), desarrollando la habilidad para transformar ideas y reconociendo oportunidades existentes para las actividades personales y profesionales; la competencia en conciencia y expresión culturales (CCEC), desarrollando la capacidad estética y creadora, para poder utilizarlas como medio de comunicación y expresión personal; y la competencia plurilingüe (CP) que le permite utilizar diferentes lenguas, orales o signadas, para comunicarse de forma apropiada y eficaz.

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.

Los saberes básicos de la materia se secuencian en cuatro bloques: el primero, «Programación»; el segundo, «Datos e información»; el tercero, «Desarrollo web» y el cuarto, «Computación física y robótica».

Durante el curso, el alumnado deberá realizar proyectos cooperativos de desarrollo de software, encuadrados en los bloques de saberes básicos 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.

Por ello, el alumnado debería desarrollar software de acuerdo con 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 el desarrollo de capacidades que les permitan, entre otras, 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.

## **1.- RELACIÓN ENTRE ELEMENTOS CURRICULARES**

### **1.1.- Criterios de evaluación**

**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.

**1.2.** Escribir programas, convenientemente estructurados y comentados, que recogen y procesan la información procedente de diferentes fuentes y generan la correspondiente salida.

**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.

**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.

**3.1.** Utilizar los lenguajes de marcado y estilos para la creación de páginas web, teniendo en cuenta aspectos relativos al diseño adaptativo.

**3.2.** Diseñar, programar y probar una aplicación web sencilla con acceso a una base de datos, utilizando un lenguaje de script.

**4.1.** Diseñar, programar y probar una aplicación que lea datos de un sensor, los procese, y como resultado, ejecute un actuador.

## **1.2.- 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.

**1.3.- Tabla 1**

<b>PROGRAMACIÓN Y COMPUTACIÓN</b>		
<b>Competencias específicas</b>	<b>Criterios de evaluación</b>	<b>Saberes básicos</b>
<p>1. Buscar y seleccionar la información adecuada proveniente de diversas fuentes, de manera crítica y segura, aplicando procesos de investigación, métodos de análisis de productos y experimentando con herramientas de simulación, para definir problemas tecnológicos e iniciar procesos de creación de soluciones a partir de la información obtenida.</p> <p>CCL1, CCL3, CP2, STEM3, CD2, CD3, CD5, CPSAA3.1, CPSAA3.2, CE1, CE3.</p>	1.1	PRYC.2.A.3.1. PRYC.2.A.3.2. PRYC.2.A.3.3. PRYC.2.A.3.4
	1.2	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.
	1.3	PRYC.2.A.3.5. PRYC.2.A.3.6. PRYC.2.A.3.7. PRYC.2.A.3.8
<p>2. Recopilar y procesar datos que ayuden en la resolución de un problema, y analizar cómo su almacenamiento, transmisión y presentación se benefician de la manipulación computacional.</p> <p>CCL1, STEM1, STEM2, STEM4, CD5, CPSAA5, CE3</p>	2.1	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>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>CCL1, CCL3, CP2, STEM3, CD2, CD3, CD5, CE1, CE3</p>	3.1	PRYC.2.C.1.1. PRYC.2.C.1.2. PRYC.2.C.1.3. PRYC.2.C.1.4.
	3.2	PRYC.2.C.2.1. PRYC.2.C.2.2. PRYC.2.C.2.3. PRYC.2.C.2.4.
<p>4. Explorar la computación física, construyendo un sistema hardware y software que interactúe con el mediofísico, detectando y respondiendo a cambios en el mundo real, para comprender las diferencias entre los mundos digital y analógico.</p> <p>CCL1, CCL3, CP2, STEM3, CD2, CD3, CD5, CE1, CE3</p>	4.1	PRYC.2.D.1.1. PRYC.2.D.1.2. PRYC.2.D.1.3. PRYC.2.D.2.1

## 2.- EVALUACIÓN

SITUACIONES DE APRENDIZAJE	INSTRUMENTOS DE OBSERVACIÓN	CRITERIOS DE EVALUACIÓN						
		1.1	1.2	1.3	2.1	3.1	3.2	4.1
SITUACIÓN APRENDIZAJE 1	IO 1	X	X	X				
	IO 2				X			
SITUACIÓN APRENDIZAJE 2	IO 3	X	X	X	X			
	IO 4	X	X	X	X			
PRIMER TRIMESTRE		Media	Media	Media	Media	Media	Media	Media
SITUACIÓN APRENDIZAJE 3	IO 1					X		
	IO 2						X	
	IO 3					X	X	
	IO 4					X	X	
SEGUNDO TRIMESTRE		Media	Media	Media	Media	Media	Media	Media
SITUACIÓN APRENDIZAJE 4	IO 1							X
	IO 2							X
	IO 3							X
	IO 4							X
TERCER TRIMESTRE		Media	Media	Media	Media	Media	Media	Media
<b>FINAL</b>		Nota Final	Nota Final	Nota Final	Nota Final	Nota Final	Nota Final	Nota Final

### **INSTRUMENTOS DE OBSERVACIÓN Y PONDERACIÓN:**

IO1: PRUEBA PUNTUABLE 1 30%

IO2: PRUEBA PUNTUABLE 2 30%

IO3: TAREAS 30%

IO4: OTRAS ACTIVIDADES RELATIVAS A LA SITUACIÓN DE APRENDIZAJE 10%

**3.- TEMPORALIZACIÓN**

<b>UNIDAD</b>	<b>Nº SESIONES</b>
<b>SITUACIÓN DE APRENDIZAJE 1: Programación</b>	18
<b>SITUACIÓN DE APRENDIZAJE 2: Datos e Información</b>	10
<b>SITUACIÓN DE APRENDIZAJE 3: Desarrollo web</b>	20
<b>SITUACIÓN DE APRENDIZAJE 4: Computación física y Robótica</b>	17
<b>TOTAL</b>	<b>65</b>

## 4.- SITUACIONES DE APRENDIZAJE

SITUACIÓN DE APRENDIZAJE 1: PROGRAMACIÓN	
<p><b>Descripción:</b> 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><b>Descriptor:</b> CCL1, CCL3, CP2, STEM3, CD2, CD3, CD5, CPSAA3.1, CPSAA3.2, CE1, CE3.</p>	<p><b>Nº sesiones:</b> 18</p>
Contenidos	Metodología
<p>1. Lenguajes de programación.</p> <p>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</p> <p>1.2. Estructuras de control condicionales e iterativas. Estructuras de datos</p> <p>1.3. Funciones y reutilización de código. Manipulación de archivos.</p> <p>2. Orientación a objetos.</p> <p>2.1. Clases, objetos y constructores. Sobrecarga, encapsulamiento y ocultación.</p> <p>2.2. Herencia. Subclases y superclases. Interfaces. Polimorfismo.</p> <p>3. Ciclo de vida del software.</p> <p>3.1. Metodologías de desarrollo de software.</p> <p>3.2. Enfoque Top-Down, fragmentación de problemas y algoritmos.</p> <p>3.3. Pseudocódigo y diagramas de flujo.</p> <p>3.4. Desarrollo iterativo.</p> <p>3.5. Entornos de desarrollo integrado.</p> <p>3.6. Análisis, Diseño, Programación y Pruebas. Depuración.</p> <p>3.7. Control de versiones.</p> <p>3.8. Trabajo en equipo.</p>	<p>- Clases teóricas.</p> <p>- Clases prácticas.</p>
<p><b>Observaciones:</b></p>	

<b>SITUACIÓN DE APRENDIZAJE 2: DATOS E INFORMACIÓN</b>	
<b>Descripción:</b> 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	
<b>Descriptor:</b> CCL1, STEM1, STEM2, STEM4, CD5, CPSAA5, CE3	<b>Nº sesiones:</b> 10
<b>Contenidos</b>	<b>Metodología</b>
<p>1. Bases de datos relacionales.</p> <p>1.1. Sistemas gestores de bases de datos. Ventajas con respecto a los archivos.</p> <p>1.2. Diseño de bases de datos relacionales. Diagramas entidad-relación, esquema relacional y normalización.</p> <p>1.3. Creación y manipulación de bases de datos relacionales. Comandos básicos de SQL: create, insert, delete, select, update.</p> <p>2. Big data.</p> <p>2.1. Volumen y variedad de datos. Datos estructurados, no estructurados y semiestructurados.</p> <p>2.2. Introducción a las bases de datos NoSQL.</p>	<p>- Clases teóricas.</p> <p>- Clases prácticas.</p>
<b>Observaciones:</b>	

<b>SITUACIÓN DE APRENDIZAJE 3: DESARROLLO WEB</b>	
<p><b>Descripción:</b> 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><b>Descriptor:</b></p> <p>CCL1, CCL3, CP2, STEM3, CD2, CD3, CD5, CE1, CE3</p>	<p><b>Nº sesiones:</b></p> <p>20</p>
<b>Contenidos</b>	<b>Metodología</b>
<p>1. Lenguajes descriptivos.</p> <p>1.1. Lenguaje de marcas de hipertexto (HTML), documentos, etiquetas, estructura, elementos, y atributos.</p> <p>1.2. Títulos, texto, listas, tablas, formularios y multimedia.</p> <p>1.3. Hojas de estilo en cascada (CSS). Reglas de estilo. Selectores. Declaraciones. Propiedades y Valores.</p> <p>1.4. El modelo de cajas. Diseño adaptativo.</p> <p>2. Lenguajes de programación.</p> <p>2.1. Visión general de los lenguajes de scripts.</p> <p>2.2. Programación en entorno cliente.</p> <p>2.3. Introducción a la programación en entorno servidor.</p> <p>2.4. Acceso a bases de datos. Interfaz de programación de aplicaciones con servicios web (REST APIs).</p>	<p>- Clases teóricas.</p> <p>- Clases prácticas.</p>
<p><b>Observaciones:</b></p>	

**SITUACIÓN DE APRENDIZAJE 4: COMPUTACIÓN Y ROBÓTICA**

**Descripción:** 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.

**Descriptor:**

CCL1, CCL3, CP2, STEM3, CD2, CD3, CD5, CE1, CE3

**Nº sesiones:**

17

**Contenidos****Metodología**

## 1. Robótica.

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

1.2. Microcontroladores, entrada/salida, sensores y actuadores.

1.3. Programación de dispositivos inteligentes.

## 2. El Internet de las Cosas.

2.1. Aplicaciones. Smart Cities.

- Clases teóricas.

- Clases prácticas.

**Observaciones:**

## **5.- MEDIDAS DE ATENCIÓN A LA DIVERSIDAD**

Recogidas en la programación de departamento.

Tras la evaluación inicial y la información aportada por el Departamento de Orientación del Centro, se detectan:

- Una alumna con Altas Capacidades. Podrían participar en un Programa de Profundización, si así se estima oportuno.