

I.E.S. SANTÍSIMA TRINIDAD

Departamento de Informática

COMPUTACIÓN Y ROBÓTICA

(Asignatura de libre configuración autonómica)

1^{er} CICLO E.S.O.

CURSO 2021/2022

PRESENTACIÓN

La computación es la disciplina dedicada al estudio, diseño y construcción de programas y sistemas informáticos, sus principios y prácticas, aplicaciones y el impacto que estas 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. La computación es el motor innovador de la sociedad del conocimiento, y se sitúa en el núcleo del denominado sector de actividad cuaternario, relacionado con la información.

Por otro lado, la robótica es un campo de investigación multidisciplinar, en la frontera entre las ciencias de la computación y la ingeniería, cuyo objetivo es el diseño, la construcción y operación de robots. Los robots son sistemas autónomos que perciben el mundo físico y actúan en consecuencia, realizando tareas al servicio de las personas. A día de hoy, se emplean de forma generalizada desarrollando trabajos en los que nos sustituyen.

Aunque resulta imposible predecir con exactitud el futuro del mundo digital, áreas de conocimiento y aplicaciones como la Inteligencia Artificial, Internet de las Cosas o los Vehículos Autónomos provocan, de forma disruptiva, cambios enormes en nuestra vida. El impacto es inmenso en todas las disciplinas, siendo el común denominador la transformación y automatización de procesos y sistemas, así como la innovación y mejora de los mismos. Por otro lado, estas tecnologías plantean cuestiones relacionadas con la privacidad, la seguridad, la legalidad o la ética, que constituyen auténticos desafíos de nuestro tiempo.

La enseñanza de la materia Computación y Robótica es estratégica para el futuro de la innovación, la investigación científica y el empleo. Descubrir los principios que rigen esta materia y ser expuestos al proceso de construcción debe promover en el alumnado vocaciones en el ámbito STEM (Science, Technology, Engineering & Maths), diseñar iniciativas que fomenten el aumento de la presencia de la mujer en estos ámbitos, romper ideas preconcebidas sobre su dificultad y dotar al alumnado de herramientas que les permitan resolver problemas complejos. Hay que señalar, además, que aprender computación permite conceptualizar y comprender mejor los sistemas digitales, transferir conocimientos entre ellos, y desarrollar una intuición sobre su funcionamiento que permite hacer un uso más productivo de los mismos.

La materia Computación y Robótica está estructurada en tres bloques de contenidos:

El primer bloque, Programación y desarrollo de software, introduce al alumnado en los lenguajes informáticos que permiten escribir programas, ya sean para equipos de sobremesa, dispositivos móviles o la web.

El segundo bloque, Computación física y robótica, trata sobre la construcción de sistemas y robots programables que interactúan con el mundo real a través de sensores, actuadores e Internet.

Por último, el tercer bloque, Datos masivos, ciberseguridad e Inteligencia Artificial, introduce los aspectos fundamentales de dichas materias y su relación con los dos bloques anteriores.

Adicionalmente, cada uno de los bloques de contenidos se subdivide en tres temáticas que se corresponderían con los contenidos de cada curso dentro de cada bloque.

En concreto, en el primer curso se tratarían los contenidos identificados con la letra A dentro de cada bloque, sobre las temáticas de “Introducción a la Programación”, “Fundamentos de la Computación Física” y “Datos Masivos”.

En segundo curso, los contenidos serían los identificados con la letra B dentro de cada bloque, sobre las temáticas de “Desarrollo Móvil”, “Internet de las Cosas” y “Ciberseguridad”.

Finalmente, en tercer curso se tratarían los contenidos identificados con la letra C dentro de cada bloque, sobre las temáticas de “Desarrollo Web”, “Robótica” e “Inteligencia Artificial”.

Cabe señalar que esta división por bloques propuesta para cada uno de los cursos es orientativa, ya que para la elección de unos contenidos u otros deberían tomarse en consideración criterios como el nivel de conocimientos previos del alumnado, su contexto socioeconómico y cultural, los recursos humanos o materiales de los que el centro pueda disponer y las necesidades sociales concretas que se detecten en el entorno de la comunidad educativa, pudiendo así trabajar las temáticas de cada bloque de manera interrelacionada. En cualquier caso, la elección de los contenidos a trabajar en cada curso debe resultar altamente motivadora para el alumnado al que vaya dirigida.

OBJETIVOS

La enseñanza de la materia Computación y Robótica tiene como finalidad el desarrollo de las siguientes capacidades:

1. Comprender el impacto que la computación y la robótica tienen en nuestra sociedad, sus aplicaciones en los diferentes ámbitos de conocimiento, beneficios, riesgos y cuestiones éticas, legales o de privacidad derivadas de su uso.
2. Desarrollar el pensamiento computacional, aprendiendo a resolver problemas con la ayuda de un ordenador u otros dispositivos de procesamiento, a saber, formularlos, a analizar información, a modelar y automatizar soluciones algorítmicas, y a evaluarlas y generalizarlas.
3. Realizar proyectos de construcción de sistemas digitales, que cubran el ciclo de vida, y se orienten preferentemente al desarrollo social y a la sostenibilidad, reaccionando a situaciones que se produzcan en su entorno y solucionando problemas del mundo real de una forma creativa.
4. Integrarse en un equipo de trabajo, colaborando y comunicándose de forma adecuada para conseguir un objetivo común, fomentando habilidades como la capacidad de resolución de conflictos y de llegar a acuerdos.
5. Producir programas informáticos plenamente funcionales utilizando las principales estructuras de un lenguaje de programación, describiendo cómo los programas implementan algoritmos y evaluando su corrección.
6. Crear aplicaciones web sencillas utilizando las librerías, frameworks o entornos de desarrollo integrado que faciliten las diferentes fases del ciclo de vida, tanto del interfaz gráfico de usuario como de la lógica computacional.
7. Comprender los principios del desarrollo móvil, creando aplicaciones sencillas y usando entornos de desarrollo integrados de trabajo online mediante lenguajes de bloques, diseñando interfaces e instalando el resultado en terminales móviles.
8. Construir sistemas de computación físicos sencillos, que conectados a Internet, generen e intercambien datos con otros dispositivos, reconociendo cuestiones relativas a la seguridad y la privacidad de los usuarios.
9. Construir sistemas robóticos sencillos, que perciban su entorno y respondan a él de forma autónoma para conseguir un objetivo, comprendiendo los principios básicos de ingeniería sobre los que se basan y reconociendo las diferentes tecnologías empleadas.
10. 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.
11. Usar aplicaciones informáticas de forma segura, responsable y respetuosa, protegiendo la identidad online y la privacidad, reconociendo contenido, contactos o conductas inapropiadas y sabiendo cómo informar al respecto.
12. Entender qué es la Inteligencia Artificial y cómo nos ayuda a mejorar nuestra comprensión del mundo, conociendo los algoritmos y técnicas empleadas en el aprendizaje automático de las máquinas, reconociendo usos en nuestra vida diaria.

COMPETENCIAS

En el aula, la competencia en comunicación lingüística (CCL) se fomentará mediante la interacción respetuosa con otros interlocutores en el trabajo en equipo, las presentaciones en público de sus creaciones y propuestas, la lectura de textos en múltiples modalidades, formatos y soportes, la redacción de documentación acerca de sus proyectos o la creación de narraciones digitales interactivas e inteligentes. Por otro lado, el dominio de los lenguajes de programación, que disponen de su propia sintaxis y semántica, contribuye especialmente a la adquisición de esta competencia.

La competencia matemática y competencias básicas en ciencia y tecnología (CMCT) se trabaja aplicando las herramientas del razonamiento matemático y los métodos propios de la racionalidad científica al diseño, implementación y prueba de los sistemas tecnológicos construidos. Además, la creación de programas que solucionen problemas de forma secuencial, iterativa, organizada y estructurada facilita el desarrollo del pensamiento matemático y computacional.

Es evidente la contribución de esta materia al desarrollo de la competencia digital (CD), a través del manejo de software para el tratamiento de la información, la utilización de herramientas de simulación de procesos tecnológicos o la programación de soluciones a problemas planteados, fomentando el uso creativo, crítico y seguro de las tecnologías de la información y comunicación.

La naturaleza de las tecnologías utilizadas, que evolucionan y cambian de manera rápida y vertiginosa, implica que el alumnado deba moverse en procesos constantes de investigación y evaluación de las nuevas herramientas y recursos y le obliga a la resolución de problemas complejos con los que no está familiarizado, desarrollando así la habilidad para iniciar, organizar y persistir en el aprendizaje y, por tanto, la competencia aprender a aprender (CAA).

Computación y Robótica contribuye también a la adquisición de las competencias sociales y cívicas (CSC), ya que el objetivo de la misma es la unión del aprendizaje con el compromiso social, a través de la valoración de los aspectos éticos relacionados con el impacto de la tecnología y el fomento de las relaciones con la sociedad civil. En este sentido, el alumnado desarrolla la capacidad para interpretar fenómenos y problemas sociales y para trabajar en equipo de forma autónoma y en colaboración continua con sus compañeros y compañeras, construyendo y compartiendo el conocimiento, llegando a acuerdos sobre las responsabilidades de cada uno y valorando el impacto de sus creaciones.

La identificación de un problema en el entorno para buscar soluciones de forma imaginativa, la planificación y la organización del trabajo hasta llegar a crear un prototipo o incluso un producto para resolverlo y la evaluación posterior de los resultados son procesos que fomentan en el alumnado el sentido de iniciativa y espíritu emprendedor (SIEP), al desarrollar su habilidad para transformar ideas en acciones y reconocer oportunidades existentes para la actividad personal y social.

Esta materia contribuye a la adquisición de la competencia conciencia y expresiones culturales (CEC), ya que el diseño de interfaces para los prototipos y productos tiene un papel determinante, lo que permite que el alumnado utilice las posibilidades que esta tecnología ofrece como medio de comunicación y herramienta de expresión personal, cultural y artística.

Finalmente, Computación y Robótica tiene un ámbito de aplicación multidisciplinar, de forma que los elementos transversales del currículo se pueden integrar como objetos de los sistemas a desarrollar. En el aula se debe, prioritariamente, promover modelos de utilidad social y desarrollo sostenible, fomentar la igualdad real y efectiva de géneros; incentivar una utilización crítica, responsable, segura y autocontrolada en el uso de las tecnologías informáticas y de las comunicaciones; crear un clima de respeto, convivencia y tolerancia en el uso de medios de comunicación electrónicos, prestando especial atención a cualquier forma de acoso, rechazo o violencia; procurar la utilización de herramientas de software libre; y minimizar el riesgo de brecha digital.

CONTENIDOS, CRITERIOS DE EVALUACIÓN, COMPETENCIAS Y ESTÁNDARES EVALUABLES

NIVEL A: Primer año en el que se imparte 1º ESO (Orientativo)		
Contenidos	Criterios de Evaluación Competencias Clave	Estándares de Aprendizaje
Bloque 1. Programación y desarrollo de software		
A. Introducción a la programación. Lenguajes visuales. Lenguajes de bloques. Secuencias de	1. Entender cómo funciona internamente un programa informático, la manera de elaborarlo	1.1. Identifica los principales tipos de instrucciones que componen un programa informático.

<p>instrucciones. Eventos. Integración de gráficos y sonido. Verdadero o falso. Decisiones. Datos y operaciones. Tareas repetitivas. Interacción con el usuario. Estructuras de datos. Azar. Ingeniería de software. Análisis y diseño. Modularización de pruebas. Parametrización.</p>	<p>y sus principales componentes. CCL, CMCT, CD, CAA. 2. Resolver la variedad de problemas que se presentan cuando se desarrolla una pieza de software y generalizar las soluciones. CCL, CMCT, CD, CAA, CSC, SIEP. 3. Realizar el ciclo de vida completo del desarrollo de una aplicación: análisis, diseño, programación y pruebas. CCL, CMCT, CD, CAA, CSC, SIEP, CEC. 4. Trabajar en equipo en el proyecto de construcción de una aplicación multimedia sencilla, colaborando y comunicándose de forma adecuada. CCL, CD, CAA, CSC, SIEP.</p>	<p>1.2. Utiliza datos y operaciones adecuadas a cada problema concreto. 1.3. Identifica diferentes herramientas utilizadas en la creación de aplicaciones. 2.1. Descompone problemas complejos en otros más pequeños e integra sus soluciones para dar respuesta al original. 2.2 Identifica similitudes entre problemas y reutiliza las soluciones. 2.3. Utiliza la creatividad basada en el pensamiento computacional para resolver problemas. 3.1. Analiza los requerimientos de la aplicación y realiza un diseño básico que responda a las necesidades del usuario. 3.2. Desarrolla el código de una aplicación en base a un diseño previo. 3.3. Elabora y ejecuta las pruebas del código desarrollado y de la usabilidad de la aplicación. 4.1. Explica las decisiones tomadas en equipo, en cuanto a la organización y planificación del trabajo. 4.2. Expresa sus ideas de forma asertiva, haciendo aportaciones al grupo y valorando las ideas de los demás.</p>
---	---	---

Bloque 2. Computación física y robótica

<p>A. Fundamentos de la computación física. Microcontroladores. Sistemas de computación. Aplicaciones e impacto. Hardware y software. Tipos. Productos Open-Source. Modelo Entrada - Procesamiento - Salida. Componentes: procesador, memoria, almacenamiento y periféricos. Programas e instrucciones. Ciclo de instrucción: fetch-decode-execute. Programación de microcontroladores con lenguajes visuales. IDEs. Depuración. Interconexión de microcontroladores. Pines de Entrada/Salida (GPIO). Protoboards. Seguridad eléctrica. Alimentación con baterías. Programación de sensores y actuadores. Lectura y escritura de señales analógicas y digitales. Entradas: pulsadores, sensores de luz, movimiento, temperatura,</p>	<p>1. Comprender el funcionamiento de los sistemas de computación física, sus componentes y principales características. CCL, CMCT, CD, CAA. 2. Reconocer el papel de la computación en nuestra sociedad. CSC, SIEP, CEC. 3. Ser capaz de construir un sistema de computación que interactúe con el mundo físico en el contexto de un problema del mundo real. CCL, CMCT, CD, CAA, CSC, SIEP, CEC. 4. Trabajar en equipo en el proyecto de construcción de un sistema sencillo de computación física, colaborando y comunicándose de forma adecuada. CCL, CD, CAA, CSC, SIEP.</p>	<p>1.1. Explica qué elementos hardware y software componen los sistemas de computación. 1.2. Describe cómo se ejecutan las instrucciones de los programas, y se manipulan los datos. 1.3. Identifica sensores y actuadores en relación a sus características y funcionamiento. 2.1. Describe aplicaciones de la computación en diferentes áreas de conocimiento. 2.2. Explica beneficios y riesgos derivados de sus aplicaciones. 3.1. Analiza los requisitos y diseña un sistema de computación física, seleccionando sus componentes. 3.2. Escribe y depura el software de control de un microcontrolador con un lenguaje de programación visual, dado el diseño de un sistema físico sencillo. 3.3. Realiza, de manera segura, el montaje e interconexión de los componentes de un sistema. 3.4. Prueba un sistema de computación</p>
--	---	--

<p>humedad, etc. Salidas: leds, leds RGB, zumbadores, altavoces, etc. Wearables y E-Textiles.</p>		<p>física en base a los requisitos del mismo y lo evalúa frente a otras alternativas. 4.1. Explica las decisiones tomadas en equipo, en cuanto a la organización y planificación del trabajo. 4.2. Expresa sus ideas de forma asertiva, haciendo aportaciones al grupo y valorando las ideas de los demás.</p>
---	--	--

Bloque 3. Datos masivos, ciberseguridad e inteligencia artificial

<p>A. Datos masivos. Big data. Características. Volumen de datos generados. Visualización, transporte y almacenaje de los datos Recogida y análisis de datos. Generación de nuevos datos. Entrada y salida de datos de los dispositivos y las apps. Periodismo de datos. Data scraping.</p>	<p>1. Conocer la naturaleza de las distintas tipologías de datos siendo conscientes de la cantidad de datos generados hoy en día; analizarlos, visualizarlos y compararlos. CCL, CMCT, CD, CAA, CSC, SIEP. 2. Comprender y utilizar el periodismo de datos. CCL, CMCT, CD. 3. Entender y distinguir los dispositivos de una ciudad inteligente. CMCT, CD, CSC.</p>	<p>1.1. Distingue, clasifica y analiza datos cuantitativos y cualitativos, así como sus metadatos. 1.2 Describe qué son el volumen y la velocidad de los datos, dentro de la gran variedad de datos existente, y comprueba la veracidad de los mismos. 1.3. Utiliza herramientas de visualización de datos para analizarlos y compararlos. 2.1. Busca y analiza datos en Internet, identificando los más relevantes y fiables. 2.2. Emplea de forma adecuada herramientas de extracción de datos, para representarlos de una forma comprensible y visual. 3.1. Identifica la relación entre los dispositivos, las apps y los sensores, identificando el flujo de datos entre ellos. 3.2. Conoce las repercusiones de la aceptación de condiciones a la hora de usar una app. 3.3. Usa procedimientos para proteger sus datos frente a las apps.</p>
---	--	---

NIVEL B: Primer año en el que se imparte 2º ESO (Orientativo)

Contenidos	Criterios de Evaluación Competencias Clave	Estándares de Aprendizaje
Bloque 1. Programación y desarrollo de software		
<p>B. Desarrollo móvil. IDEs de lenguajes de bloques para móviles. Programación orientada a eventos. Definición de evento. Generadores de eventos: los sensores. E/S, captura de eventos y su respuesta. Bloques de control: condicionales y bucles. Almacenamiento del estado: variables. Diseño de interfaces: la GUI.</p>	<p>1. Entender el funcionamiento interno de las aplicaciones móviles, y cómo se construyen. CCL, CMCT, CD, CAA. 2. Resolver la variedad de problemas que se presentan cuando se desarrolla una aplicación móvil, y generalizar las soluciones. CCL, CMCT, CD, CAA, CSC, SIEP. 3. Realizar el ciclo de vida completo del desarrollo de una aplicación</p>	<p>1.1. Describe los principales componentes de una aplicación móvil. 1.2. Identifica diferentes herramientas utilizadas en la creación de aplicaciones móviles. 2.1. Descompone problemas complejos en otros más pequeños e integra sus soluciones para dar respuesta al original. 2.2. Identifica similitudes entre</p>

<p>Elementos de organización espacial en la pantalla. Los gestores de ubicación. Componentes básicos de una GUI: botones, etiquetas, cajas de edición de texto, imágenes, lienzo. Las pantallas. Comunicación entre las distintas pantallas. Ingeniería de software. Análisis y diseño. Programación. Modularización de pruebas. Parametrización.</p>	<p>móvil: análisis, diseño, programación, pruebas. CCL, CMCT, CD, CAA, CSC, SIEP, CEC. 4. Trabajar en equipo en el proyecto de construcción de una aplicación móvil sencilla, colaborando y comunicándose de forma adecuada. CCL, CD, CAA, CSC, SIEP.</p>	<p>problemas y reutiliza las soluciones. 2.3. Realiza un análisis comparativo de aplicaciones móviles con sus equivalentes de escritorio. 2.4. Utiliza la creatividad basada en el pensamiento computacional para resolver problemas. 3.1. Analiza los requerimientos de una aplicación móvil sencilla. 3.2. Realiza un diseño básico de la lógica e interfaz de usuario que responda a los requerimientos. 3.3. Desarrolla el código de una aplicación móvil en base a un diseño previo. 3.4. Elabora y ejecuta, en dispositivos físicos, las pruebas del código desarrollado y de la usabilidad de la aplicación. 4.1. Explica las decisiones tomadas en equipo, en cuanto a la organización y planificación del trabajo. 4.2. Expresa sus ideas de forma asertiva, haciendo aportaciones al grupo y valorando las ideas de los demás</p>
<p>Bloque 2. Computación física y robótica</p>		
<p>B. Internet de las Cosas. Definición. Historia. Ley de Moore. Aplicaciones. Seguridad, privacidad y legalidad. Componentes: dispositivos con sensores y actuadores, red y conectividad, datos e interfaz de usuario. Modelo de conexión de dispositivo a dispositivo. Conexión BLE. Aplicaciones móviles IoT. Internet de las Cosas y la nube. Internet. Computación en la nube. Servicios. Modelo de conexión dispositivo a la nube. Plataformas. Gateways. Web Of Things. Smart Cities. Futuro IoT.</p>	<p>1. Comprender el funcionamiento de Internet de las Cosas, sus componentes y principales características. CCL, CMCT, CD, CAA. 2. Conocer el impacto de Internet de las Cosas en nuestra sociedad, haciendo un uso seguro de estos dispositivos. CSC, SIEP, CEC. 3. Ser capaz de construir un sistema de computación IoT, que conectado a Internet, genere e intercambie datos, en el contexto de un problema del mundo real. CCL, CMCT, CD, CAA, CSC, SIEP, CEC. 4. Trabajar en equipo en el proyecto de construcción de un sistema de computación IoT, colaborando y comunicándose de forma adecuada. CCL, CD, CAA, CSC, SIEP.</p>	<p>1.1. Explica qué es Internet de las Cosas y el funcionamiento general de los dispositivos IoT. 1.2. Identifica los diferentes elementos hardware y software de los sistemas IoT en relación a sus características y funcionamiento. 2.1. Identifica dispositivos IoT y sus aplicaciones en múltiples ámbitos. 2.2. Describe cuestiones referentes a la privacidad, seguridad y legalidad de su funcionamiento. 2.3. Configura dispositivos IoT mediante aplicaciones móviles y hace uso de ajustes de privacidad y seguridad. 3.1. Explica los requisitos de un sistema de computación IoT sencillo, analizando su descripción en texto y lo relaciona con problemas y soluciones similares. 3.2. Diseña un sistema IoT, dados unos requisitos, seleccionando sus componentes. 3.3. Escribe y depura el software de control de un microcontrolador con un lenguaje de programación visual, dado el diseño de un sistema IoT sencillo. 3.4. Realiza, de manera segura, el montaje, la configuración e interconexión de los componentes</p>

		<p>de un sistema IoT.</p> <p>3.5. Prueba un sistema IoT en base a los requisitos del mismo y lo evalúa frente a otras alternativas.</p> <p>4.1. Explica las decisiones tomadas en equipo, en cuanto a la organización y planificación del trabajo.</p> <p>4.2. Expresa sus ideas de forma asertiva, haciendo aportaciones al grupo y valorando las ideas de los demás.</p>
--	--	--

Bloque 3. Datos masivos, ciberseguridad e inteligencia artificial

<p>B. Ciberseguridad.</p> <p>Seguridad en Internet. Seguridad activa y pasiva.</p> <p>Exposición en el uso de sistemas. Malware y antimalware.</p> <p>Exposición de los usuarios: suplantación de identidad, ciberacoso, etc. Conexión a redes WIFI.</p> <p>Usos en la interacción de plataformas virtuales.</p> <p>Ley de propiedad intelectual.</p> <p>Materiales libres o propietarios en la web.</p>	<p>1. Conocer los criterios de seguridad y ser responsable a la hora de utilizar los servicios de intercambio y publicación de información en Internet. CD, CAA, CSC, CEC.</p> <p>2. Entender y reconocer los derechos de autor de los materiales que usamos en Internet. CCL, CD, CSC, CEC</p> <p>3. Seguir, conocer y adoptar conductas de seguridad y hábitos que permitan la protección del individuo en su interacción en la red. CD, CAA, CSC, CEC.</p>	<p>1.1. Utiliza Internet de forma responsable, respetando la propiedad intelectual en el intercambio de información</p> <p>2.1. Consulta distintas fuentes y utiliza el servicio web, dando importancia a la identidad digital</p> <p>2.2. Diferencia los materiales sujetos a derechos de autor frente a los de libre distribución.</p> <p>3.1. Aplica hábitos correctos en plataformas virtuales y emplea contraseñas seguras.</p> <p>3.2. Diferencia de forma correcta el intercambio de información seguro y no seguro.</p> <p>3.3. Identifica y conoce los tipos de fraude del servicio web.</p>
--	---	---

NIVEL C: Primer año en el que se imparte 3º ESO (Orientativo)

Contenidos	Criterios de Evaluación Competencias Clave	Estándares de Aprendizaje
Bloque 1. Programación y desarrollo de software		
<p>C. Desarrollo web.</p> <p>Páginas web. Estructura básica.</p> <p>Servidores web.</p> <p>Herramientas para desarrolladores.</p> <p>Lenguajes para la web. HTML.</p> <p>Scripts. Canvas. Sprites. Añadiendo gráficos. Sonido.</p> <p>Variables, constantes, cadenas y números.</p> <p>Operadores.</p> <p>Condicionales. Bucles.</p> <p>Funciones.</p> <p>El bucle del juego.</p> <p>Objetos. Animación de los gráficos.</p>	<p>1. Entender el funcionamiento interno de las páginas web y las aplicaciones web, y cómo se construyen. CCL, CMCT, CD, CAA.</p> <p>2. Resolver la variedad de problemas que se presentan cuando se desarrolla una aplicación web, y generalizar las soluciones. CCL, CMCT, CD, CAA, CSC, SIEP.</p> <p>3. Realizar el ciclo de vida completo del desarrollo de una aplicación web: análisis,</p>	<p>1.1. Describe los principales elementos de una página web y de una aplicación web.</p> <p>1.2. Identifica diferentes herramientas utilizadas en la creación de páginas y aplicaciones web.</p> <p>2.1. Descompone problemas complejos en otros más pequeños e integra sus soluciones para dar respuesta al original.</p> <p>2.2. Identifica similitudes entre problemas y reutiliza</p>

<p>Eventos. Interacción con el usuario. Ingeniería de software. Análisis y diseño. Programación. Modularización de pruebas. Parametrización.</p>	<p>diseño, programación, pruebas. CCL, CMCT, CD, CAA, CSC, SIEP, CEC. 4. Trabajar en equipo en el proyecto de construcción de una aplicación web sencilla, colaborando y comunicándose de forma adecuada. CCL, CD, CAA, CSC, SIEP.</p>	<p>las soluciones. 2.3. Realiza un análisis comparativo de aplicaciones web con sus equivalentes móviles o de escritorio. 2.4. Utiliza la creatividad basada en el pensamiento computacional para resolver problemas. 3.1. Analiza los requerimientos de una aplicación web sencilla. 3.2. Realiza un diseño básico de la lógica e interfaz de usuario que responda a los requerimientos. 3.3. Desarrolla el código de una aplicación web en base a un diseño previo. 3.4. Elabora y ejecuta las pruebas del código desarrollado y de la usabilidad de la aplicación. 4.1. Explica las decisiones tomadas en equipo, en cuanto a la organización y planificación del trabajo. 4.2. Expresa sus ideas de forma asertiva, haciendo aportaciones al grupo y valorando las ideas de los demás.</p>
<p>Bloque 2. Computación física y robótica</p>		
<p>C. Robótica. Definición de robot. Historia. Aplicaciones. Leyes de la robótica. Ética. Componentes: sensores, efectores y actuadores, sistema de control y alimentación. Mecanismos de locomoción y manipulación: ruedas, patas, cadenas, hélices, pinzas. Entradas: sensores de distancia, sensores de sonido, sensores luminosos, acelerómetro y magnetómetro. Salidas: motores dc (servomotores y motores paso a paso). Programación con lenguajes de texto de microprocesadores. Lenguajes de alto y bajo nivel. Código máquina. Operaciones de lectura y escritura con sensores y actuadores. Operaciones con archivos. Diseño y construcción de robots móviles y/o estacionarios. Robótica e Inteligencia Artificial.</p>	<p>1. Comprender los principios de ingeniería en los que se basan los robots, su funcionamiento, componentes y características. CCL, CMCT, CD, CAA. 2. Comprender el impacto presente y futuro de la robótica en nuestra sociedad. CSC, SIEP, CEC. 3. Ser capaz de construir un sistema robótico móvil, en el contexto de un problema del mundo real. CCL, CMCT, CD, CAA, CSC, SIEP, CEC. 4. Trabajar en equipo en el proyecto de construcción de un sistema robótico, colaborando y comunicándose de forma adecuada. CCL, CD, CAA, CSC, SIEP.</p>	<p>1.1. Explica qué es un robot. 1.2. Describe el funcionamiento general de un robot e identifica las tecnologías vinculadas. 1.3. Identifica los diferentes elementos de un robot en relación a sus características y funcionamiento. 2.1. Clasifica robots en base a su campo de aplicación y sus características. 2.2. Describe cuestiones éticas vinculadas al comportamiento de los robots. 2.3. Explica beneficios y riesgos derivados del uso de robots. 3.1. Describe los requisitos de un sistema robótico sencillo, analizando su descripción en texto y lo relaciona con problemas y soluciones similares. 3.2. Diseña un sistema robótico móvil, dados unos requisitos, seleccionando sus componentes. 3.3. Escribe el software de control de un sistema robótico sencillo, en</p>

El futuro de la robótica		<p>base al diseño, con un lenguaje de programación textual y depura el código.</p> <p>3.4. Realiza, de manera segura, el montaje, la configuración e interconexión de los componentes de un sistema robótico.</p> <p>3.5. Prueba un sistema robótico en base a los requisitos del mismo y lo evalúa frente a otras alternativas.</p> <p>4.1. Explica las decisiones tomadas en equipo, en cuanto a la organización y planificación del trabajo.</p> <p>4.2. Expresa sus ideas de forma asertiva, haciendo aportaciones al grupo y valorando las ideas de los demás.</p>
Bloque 3. Datos masivos, ciberseguridad e inteligencia artificial		
<p>C. Inteligencia Artificial. Definición. Historia. El test de Turing. Aplicaciones. Impacto. Ética y responsabilidad social de los algoritmos. Beneficios y posibles riesgos. Agentes inteligentes simples. Síntesis y reconocimiento de voz. Aprendizaje automático. Datos masivos. Tipos de aprendizaje. Servicios de Inteligencia Artificial en la nube. APIs. Reconocimiento y clasificación de imágenes. Entrenamiento. Reconocimiento facial. Reconocimiento de texto. Análisis de sentimiento. Traducción.</p>	<p>1. Comprender los principios básicos de funcionamiento de los agentes inteligentes y de las técnicas de aprendizaje automático. CCL, CMCT, CD, CAA.</p> <p>2. Conocer el impacto de la Inteligencia Artificial en nuestra sociedad, y las posibilidades que ofrece para mejorar nuestra comprensión del mundo. CSC, SIEP, CEC.</p> <p>3. Ser capaz de construir una aplicación sencilla que incorpore alguna funcionalidad enmarcada dentro de la Inteligencia Artificial. CCL, CMCT, CD, CAA, CSC, SIEP, CEC.</p>	<p>1.1. Explica qué es la Inteligencia Artificial.</p> <p>1.2. Describe el funcionamiento general de un agente inteligente.</p> <p>1.3. Identifica diferentes tipos de aprendizaje.</p> <p>2.1. Identifica aplicaciones de la Inteligencia Artificial y su uso en nuestro día a día.</p> <p>2.2. Describe cuestiones éticas vinculadas a la Inteligencia Artificial.</p> <p>3.1. Escribe el código de una aplicación que incorpora alguna funcionalidad de Inteligencia Artificial, utilizando herramientas que permiten crear y probar agentes sencillos.</p> <p>3.2. Elabora y ejecuta las pruebas del código desarrollado.</p>

SECUENCIACIÓN Y TEMPORALIZACIÓN

Propuesta realizada en función del cumplimiento exacto de los elementos curriculares presentes en la legislación. Es una propuesta irreal en función del nivel de desarrollo del alumnado de 1º a 3º en ESO.

NIVEL A			
Evaluación	Unidades Didácticas	Bloques de contenidos	Horas
1ª	Evaluación Inicial		2
1ª	UD1. SCRATCH	Bloque 1. Programación y desarrollo de software	24
2ª	UD2. ARDUINO con bloques	Bloque 2. Computación física y robótica	22
3ª	UD3. Processig	Bloque 3. Datos masivos, ciberseguridad e Inteligencia Artificial	20

NIVEL B			
Evaluación	Unidades Didácticas	Bloques de contenidos	Horas
1ª	Evaluación Inicial		2
1ª	UD1. APP Inventor	Bloque 1. Programación y desarrollo de software	24
2ª	UD2. ARDUINO controlado por móvil	Bloque 2. Computación física y robótica	22
3ª	UD3. Ciberseguridad	Bloque 3. Datos masivos, ciberseguridad e Inteligencia Artificial	20

NIVEL C			
Evaluación	Unidades Didácticas	Bloques de contenidos	Horas
1ª	Evaluación Inicial		2
1ª	UD1. Desarrollo de aplicaciones web	Bloque 1. Programación y desarrollo de software	24
2ª	UD2. Robot con ARDUINO	Bloque 2. Computación física y robótica	22
3ª	UD3. Inteligencia Artificial	Bloque 3. Datos masivos, ciberseguridad e Inteligencia Artificial	20

La legislación permite seleccionar los contenidos entre los distintos niveles en función de las características del alumnado de ESO, y siempre en función de las necesidades e intereses de los alumnos. En base a estas premisas, se realiza la siguiente propuesta, mucho más realista y motivadora para el alumnado.

NIVEL A			
Evaluación	Unidades Didácticas	Bloques de contenidos	Horas
1ª	Evaluación Inicial		2
1ª	UD1. Introducción a SCRATCH	Bloque 1. Programación y desarrollo de software	24
2ª	UD2. Introducción a ARDUINO con bloques en simulador.	Bloque 2. Computación física y robótica	22
3ª	UD3. Introducción a ARDUINO por bloques con elementos físicos, visualizando datos en el IDE por el puerto serie.	Bloque 3. Datos masivos, ciberseguridad e Inteligencia Artificial	20

NIVEL B			
Evaluación	Unidades Didácticas	Bloques de contenidos	Horas
1 ^a	Evaluación Inicial		2
1 ^a	UD1. Scratch avanzado	Bloque 1. Programación y desarrollo de software	24
2 ^a	UD2. Construcción de robots usando ARDUINO con bloques y código	Bloque 2. Computación física y robótica	22
3 ^a	UD3. Introducción a Processing para tratamiento de datos.	Bloque 3. Datos masivos, ciberseguridad e Inteligencia Artificial	20

NIVEL C			
Evaluación	Unidades Didácticas	Bloques de contenidos	Horas
1 ^a	Evaluación Inicial		2
1 ^a	UD1. Creación de aplicaciones con APP Inventor.	Bloque 1. Programación y desarrollo de software	20
2 ^a	UD2. Diseño 3D con Tinkercard. Impresión 3D	Bloque 1. Programación y desarrollo de software Bloque 2. Computación física y robótica	20
3 ^a	UD3. Control por móvil de robot con movimiento usando ARDUINO.	Bloque 2. Computación física y robótica	20
3 ^a	UD4. Introducción a Ciberseguridad e Inteligencia Artificial.	Bloque 3. Datos masivos, ciberseguridad e Inteligencia Artificial	8

EVALUACIÓN.

Rúbrica para cada unidad.

Indicadores de logro	
1	No asimila ni aplica los contenidos mínimos.
2	Asimila y aplica los contenidos mínimos.
3	Asimila y aplica la mayoría de los contenidos.
4	Asimila y aplica todos los contenidos correctamente.

RECUPERACIONES.

Habrá un examen de recuperación por cada trimestre, una vez concluido el mismo.

En cuanto a la calificación en junio, se tendrá en cuenta que quien haya suspendido dos o más trimestres irá con todo a la prueba final global de todo el curso, pudiendo ir con sólo una parte si sólo tiene un trimestre suspenso (después de haber hecho las recuperaciones correspondientes de cada trimestre).

A los alumnos suspensos en un trimestre, se les dará una relación con actividades correspondientes a ese trimestre para que la elaboren en casa y luego en el aula, se resuelvan los problemas que les surjan. Cambien se harán puntualmente actividades de recuerdo del trimestre anterior, en el trimestre actual.

ATENCIÓN A LA DIVERSIDAD

Uno de los retos fundamentales de la Educación Secundaria Obligatoria y post obligatoria, consiste en dar respuesta a las necesidades educativas de todo el alumnado. Es necesario ofrecer respuestas diferenciadas en función de la diversidad del alumnado. Para lograr este ajuste, pueden llevarse a cabo las siguientes medidas:

Actividades diversas y graduadas: La diversificación de actividades, por un lado permite conectar con los diferentes intereses de los alumnos y por otro lado realizarán todo tipo de actividades y no se limitarán únicamente a aquéllas que más sencillas le resulten. La diversificación de tareas a las que se les da la misma valoración aumenta la autoestima de los alumnos. El profesor tendrá que graduar las dificultades de los contenidos dentro de la unidad didáctica. A su vez, una misma actividad puede plantearse con varios grados de exigencia, trabajando con algunos alumnos sólo los contenidos mínimos previamente seleccionados que entren en ella. Entre la variada gama de actividades que pueden utilizarse para que se realice un aprendizaje efectivo y se pueda responder a la diversidad de intereses y niveles de la clase.

Actividades de desarrollo: encaminadas a adquirir los contenidos programados. Existen diferentes tipos:

- Actividades para detectar las ideas previas.
- Actividades de descubrimiento dirigido.
- Actividades de tipo autoevaluación.
- Actividades de consolidación: esquemas, mapas conceptuales, etc.
- Actividades de investigación libre.
- Realización de pequeños proyectos.
- Resolución de problemas.
- Actividades encaminadas a la búsqueda de información.

Actividades de recuperación: programadas para alumnos que no han alcanzado los conocimientos trabajados. Podrían ser muchas de las ya utilizadas descompuestas en otras más sencillas.

Actividades de ampliación: permitirán desarrollar adecuadamente las capacidades de los alumnos más aventajados. Son especialmente útiles las investigaciones libres y la resolución de problemas de papel y lápiz, con diferentes grados de dificultad. Es importante diseñarlas con un grado alto de autonomía porque permiten al profesor atender a la vez a otros alumnos que lo necesiten más.

METODOLOGÍA

- Aprendizaje activo e inclusivo

El aprendizaje debe ser activo y llevarse a cabo a través de actividades contextualizadas en el desarrollo de sistemas de computación y robóticos. Para ello, se deben emplear estrategias didácticas variadas que faciliten la atención a la diversidad, utilizando diferentes formatos y métodos en las explicaciones, trabajo de clase y tareas. Además, las actividades deben alinearse con los objetivos, tomando como referencia los conocimientos previos del alumnado.

- Aprendizaje y servicio

Es un objetivo primordial de esta materia unir el aprendizaje con el compromiso social. Combinar el aprendizaje y el servicio a la comunidad en un trabajo motivador permite mejorar nuestro entorno y formar a ciudadanos responsables. Así, podemos unir pensamiento lógico y crítico, creatividad, emprendimiento e innovación, conectándolos con los valores, las necesidades y las expectativas de nuestra sociedad. Desde un enfoque constructorista, se propone que el alumnado construya sus propios productos, prototipos o artefactos computacionales, tales como programas, simulaciones, visualizaciones, narraciones y animaciones digitales, sistemas robóticos y aplicaciones web o para dispositivos móviles, entre otros. Estas creaciones, además de conectar con los intereses del alumnado, deben dar solución a algún problema o necesidad real identificado por él mismo que le afecte de manera directa o al entorno del propio centro docente. De esta forma, se aprende interviniendo y haciendo un servicio para la comunidad educativa, lo que a su vez requiere la coordinación con entidades sociales.

- Aprendizaje basado en proyectos

El aprendizaje de sistemas de computación y/o robóticos debe estar basado en proyectos y, por ello, se recomienda realizar tres proyectos durante el curso (uno en cada trimestre). Alternativamente al desarrollo completo de un proyecto, y dependiendo de las circunstancias, se podrían proponer proyectos de ejemplo (guiados y cerrados) o bien proyectos basados en una plantilla (el alumnado implementa solo algunas partes del sistema, escribiendo bloques del código).

- Ciclo de desarrollo

El ciclo de desarrollo se debe basar en prototipos que evolucionan hacia el producto final. Este proceso se organizará en iteraciones que cubran el análisis, diseño, programación y/o montaje, pruebas, y en las que se añaden nuevas funcionalidades. Además, se deben planificar los recursos y las tareas, mantener la documentación y evaluar el trabajo propio y el del equipo. Por último, se almacenarán los archivos de los proyectos en un portfolio personal, que podría ser presentado en público.

- Resolución de problemas

La resolución de problemas se debe trabajar en clase con la práctica de diferentes técnicas y estrategias. De manera sistemática, a la hora de enfrentarnos a un problema, se tratará la recopilación de la información necesaria, el filtrado de detalles innecesarios, la descomposición en subproblemas, la reducción de la complejidad creando versiones más sencillas y la identificación de patrones o similitudes entre problemas. En cuanto a su resolución, se incidirá en la reutilización de conocimientos o soluciones existentes, su representación visual, diseño algorítmico, evaluación y prueba, refinamiento y comparación con otras alternativas en términos de eficiencia. Por último, habilidades como la persistencia y la tolerancia a la ambigüedad se pueden trabajar mediante el planteamiento de problemas abiertos.

- Análisis y diseño

La creación de modelos y representaciones es una técnica muy establecida en la disciplina porque nos permite comprender mejor el problema e idear su solución. A nivel escolar, se pueden emplear descripciones textuales de los sistemas, tablas de requisitos, diagramas de objetos y escenarios (animaciones y videojuegos), diagramas de componentes y flujos de datos (sistemas físicos y aplicaciones móviles), diagramas de interfaz de usuario (aplicaciones móviles y web), tablas de interacciones entre objetos (videojuegos), diagramas de secuencias (sistemas físicos, aplicaciones móviles y web). Adicionalmente, se podrían emplear diagramas de estado, de flujo o pseudocódigo.

- Programación

Aprender a programar se puede llevar a cabo realizando diferentes tipos de ejercicios, entre otros, ejercicios predictivos donde se pide determinar el resultado de un fragmento de código, ejercicios de esquema donde se pide completar un fragmento incompleto de código, ejercicios de Parsons donde se pide ordenar unas instrucciones desordenadas, ejercicios de escritura de trazas, ejercicios de escritura de un programa o fragmento que satisfaga una especificación y ejercicios de depuración donde se pide corregir un código o indicar las razones de un error. Estas actividades se pueden también realizar de forma escrita u oral, sin medios digitales (actividades desenchufadas).

- Sistemas físicos y robóticos

En la construcción de sistemas físicos y robóticos, se recomienda crear el diagrama esquemático, realizar la selección de componentes electrónicos y mecánicos entre los disponibles en el mercado, diseñar el objeto 3D o algunos de los componentes, montar de forma segura el sistema (debe evitarse la red eléctrica y usar pilas en su alimentación), y llevar a cabo pruebas funcionales y de usabilidad. Por otro lado, se pueden emplear simuladores que ayuden a desarrollar los sistemas de forma virtual, en caso de que se considere conveniente.

- Colaboración y comunicación

La colaboración, la comunicación, la negociación y la resolución de conflictos para conseguir un objetivo común son aprendizajes clave a lo largo de la vida. En las actividades de trabajo en equipo, se debe incidir en aspectos de coordinación, organización y autonomía, así como tratar de fomentar habilidades como la empatía o la asertividad y otras enmarcadas dentro de la educación emocional. Además, es importante que los estudiantes adquieran un nivel básico en el uso de herramientas software de productividad.

- Educación científica

La educación científica del alumnado debe enfocarse a proporcionar una visión globalizada del conocimiento. Por ello, se debe dar visibilidad a las conexiones y sinergias entre la computación y otras ramas de conocimiento como forma de divulgación científica, e incidir en cuestiones éticas de aplicaciones e investigaciones.

- Sistemas de gestión del aprendizaje online

Los entornos de aprendizaje online dinamizan la enseñanza-aprendizaje y facilitan aspectos como la interacción profesorado-alumnado, la atención personalizada y la evaluación. Por ello, se recomienda el uso generalizado de los mismos.

- Software y hardware libre

El fomento de la filosofía de hardware y software libre se debe promover priorizando el uso en el aula de programas y dispositivos de código abierto, y entenderse como una forma de cultura colaborativa.

DIDÁCTICA

El material didáctico lo proporcionará el profesor desglosado en bloques.

Entre los recursos didácticos contamos con 15 ordenadores personales (habrá 2 personas por ordenador como máximo), cañón para proyecciones, reproductor multimedia y acceso a internet.

Material específico por niveles:

Nivel A

- ✓ Ordenadores con conexión a Internet para el uso de SCRATCH, TINKERCAD para programación con ARDUINO en simulador y ArduinoBlocks para programar ARDUINO por bloques.
- ✓ Ordenador con el IDE de ARDUINO instalado para poder comunicarse con la placa ARDUINO.
- ✓ Placa de ARDUINO UNO, placa de puntos, indicadores, sensores, actuadores, elementos de protección, cableado y pilas.

Nivel B

- ✓ Ordenadores con conexión a Internet para el uso de SCRATCH, con cámara para poder realizar programas en los que el usuario pueda interactuar con el programa.
- ✓ Ordenador con el IDE de ARDUINO instalado para poder comunicarse con la placa ARDUINO.
- ✓ Placa de ARDUINO UNO, placa de puntos, indicadores, sensores, actuadores, elementos de protección, cableado y pilas para poder montar un robot.
- ✓ Ordenador con PROCESSING instalado para poder comunicarse con la placa ARDUINO y así poder tratar los datos que ésta proporcione.

Nivel C

- ✓ Ordenadores con conexión a Internet para el uso del programa APP INVENTOR, del MIT.
- ✓ Teléfono móvil de alumnos Android, para poder ejecutar las aplicaciones realizadas con APP INVENTOR.
- ✓ Ordenadores con conexión a Internet para el uso de TINKERCAD para realizar diseños 3D.
- ✓ Impresoras 3D para poder imprimir los diseños realizados con TINKERCAD.
- ✓ Placa de ARDUINO UNO, placa de puntos, indicadores, sensores, actuadores, elementos de protección, cableado y pilas para poder montar un robot.
- ✓ Ordenador con conexión a Internet para que el alumnado pueda recopilar información sobre
- ✓ Ciberseguridad e Inteligencia Artificial.

MECANISMOS PARA LA REVISIÓN, SEGUIMIENTO Y EVALUACIÓN DE LA PROGRAMACIÓN.

La finalidad de la evaluación educativa es mejorar el proceso de aprendizaje de cada alumno, el funcionamiento del grupo clase y nuestra propia práctica.

Para la revisión, seguimiento y evaluación de la programación, me baso en los siguientes ítems y compruebo concordancia entre la evolución del grupo y lo estipulado en la programación:

1. Tener en cuenta el procedimiento general, que concreto en mi programación de aula, para la evaluación de los aprendizajes de acuerdo con la programación de área.
2. Aplicar los estándares de evaluación de acuerdo con las programaciones de áreas,
3. Realizar una evaluación inicial a principio de curso, para ajustar la programación, en la que tengo en cuenta la opinión del Equipo Educativo y el Departamento de Orientación.
4. Utilizar sistemáticamente procedimientos e instrumentos variados de recogida de información (registro de observaciones, actividades del alumno)

5. Corregir y explicar -habitual y sistemáticamente- los trabajos y actividades de los alumnos y, dar pautas para la mejora de sus aprendizajes.
6. Usar estrategias y procedimientos de autoevaluación y coevaluación en grupo que favorezcan la participación de los alumnos en la evaluación.
7. Utilizar diferentes técnicas de evaluación en función de la diversidad de alumnos/as
8. Utilizar diferentes medios para informar a padres, profesores y alumnos (sesiones de evaluación, información personalizada a petición del tutor o padres, reuniones equipo educativo) de los resultados de la evaluación y/o en el momento en la que se produzca la petición.

LEGISLACIÓN.

Orden de 15 de enero de 2021, por la que se desarrolla el currículo correspondiente a la etapa de Educación Secundaria Obligatoria en la Comunidad Autónoma de Andalucía, se regulan determinados aspectos de la atención a la diversidad, se establece la ordenación de la evaluación del proceso de aprendizaje del alumnado y se determina el proceso de tránsito entre distintas etapas educativas.

