

PROGRAMACIÓN DIDÁCTICA

2º BACHILLERATO DISTANCIA DE CIENCIAS Y TECNOLOGÍA - PROGRAMACIÓN (DISTANCIA)

Programación (Optativa) - 2º Bachillerato de Ciencias y Tecnología

I.E.S. Práxedes Mateo Sagasta (26001560) 2025/2026

Fechas de comienzo y fin

Inicio aproximado: 16-09-2025

Finalización aproximada: 22-06-2026

Jefe del departamento responsable de la programación

M Carmen Sainz de Ugarte Fernández

Docentes implicados en el desarrollo de la programación

- Eugenio Ulecia Blanco
- Jorge Muro Hernández
- Juan Carlos López Gil
- Elena González Caballero
- Miguel Arraiz Fernández
- M Carmen Sainz de Ugarte Fernández

PROCEDIMIENTO PARA LA ADOPCIÓN DE MEDIDAS DE ATENCIÓN A LA DIVERSIDAD

El profesor deberá tener presente que los alumnos y alumnas tienen distintos intereses y motivaciones, así como diferentes ritmos de aprendizaje. La clase es una diversidad a la que debe dar respuestas lo más individualizadas posibles. Las medidas de atención a la diversidad, en los casos más extremos, se llevarán siempre a cabo en coordinación con el Departamento de Orientación del centro. Podemos distinguir dos niveles de actuación:

a) Adaptaciones curriculares de acceso.

b) Adaptaciones curriculares significativas.

a) Adaptaciones curriculares de acceso

La profesora o profesor realiza algunos cambios en la metodología, actividades, materiales o agrupamientos, para atender a diferencias individuales o a dificultades de aprendizaje del alumnado, que no afectan a los objetivos de la etapa ni a los contenidos mínimos.

◆ **Destinatarios:** aquellos alumnos o alumnas que presentan dificultades de aprendizaje pero que no afectan a su currículo. También para alumnos con altas capacidades o alumnos con un nivel de conocimientos superior a la media

. Pediremos ayuda y asesoramiento al Departamento de Orientación del centro para detectar a los alumnos con dificultades en sus aprendizajes, así como para recibir sus propuestas de actuación y materiales, para los casos necesarios.

◆ **Metodología:** será variada para responder a sus necesidades de aprendizaje y en función de:

- El nivel de conocimientos previos de cada alumno o alumna. •

El grado de autonomía personal.

- La identificación de las dificultades en etapas anteriores.

- Introducción de nuevos contenidos de acuerdo a sus posibilidades.

◆ **Actividades:**

- De refuerzo: de lo que saben hacer, pero deben consolidar.
- De ampliación: para adquirir conocimientos y habilidades de un nivel más avanzado que los aportados por las actividades programadas.

◆ **Propuesta de actuación:**

- Consolidar contenidos.
- Ejercitar actividades instrumentales básicas (lenguaje y matemáticas) en los contenidos de tecnología.
- Proporcionar actividades de refuerzo para superar dificultades concretas.
- Graduar las dificultades de las tareas. Partiremos de conceptos simples para conseguir logros básicos y, a partir de ellos, ampliar de acuerdo con las posibilidades de cada alumno.
- Conducir el proceso de trabajo con el nivel de ayudas necesarias, para que el propio alumno llegue a la solución.

◆ **Agrupamiento:** distribuir a estos alumnos con dificultades en equipos de trabajo heterogéneos adaptando la realización de tareas dentro del grupo a sus posibilidades.

b) Adaptaciones curriculares significativas

Estas adaptaciones deberán adecuar los objetivos, la supresión o modificación de los contenidos mínimos y el cambio en los criterios de evaluación, para lograr superar ese nuevo currículo, de acuerdo con sus posibilidades. El Departamento de Orientación del centro nos asesorará en la realización de la adaptación del currículo y nos propondrá pautas de actuación con estos alumnos, así como la conveniencia de su asistencia al aula ordinaria, en horario total o parcial, de acuerdo con las necesidades educativas especiales de cada alumno.

◆ **Destinatarios:** alumnos o alumnas que presentan necesidades educativas especiales.

- Alumnos que presentan limitaciones de tipo físico, psíquico o sensorial.
- Alumnos con un historial escolar y social que ha producido limitaciones tan significativas en sus aprendizajes, asociadas a desinterés y desmotivación, que impiden la adquisición de nuevos contenidos.
- Alumnos inmigrantes que desconozcan el idioma.

◆ **Punto de partida inicial:** los alumnos tendrán un diagnóstico de sus necesidades especiales, realizado por los profesionales correspondientes, quienes orientarán al profesor de Tecnología en la realización de su adaptación curricular.

◆ **Metodología:** debe ser variada a la hora de responder a las necesidades de aprendizaje y en función de:

- Las necesidades educativas especiales de cada alumno o alumna.
- La metodología utilizada en otros cursos.
- El grado de autonomía personal.

- El nivel de conocimientos previos de cada uno.
- Introducir cambios en su currículo según supere, o no, objetivos.

◆ **Actividades:**

- Adecuadas a su adaptación curricular.
- De refuerzo de lo que sabe hacer, pero debe consolidar.
- Actualizadas, de acuerdo a sus avances y retrocesos.
- De ampliación: de lo que puede hacer y no hace por falta de aprendizajes básicos.
- Graduar las dificultades de las tareas. Partiremos de conceptos simples para conseguir logros básicos y, a partir de ellos, ampliar de acuerdo con las posibilidades de cada alumno.
- Conducir el proceso de trabajo con el nivel de ayudas necesarias, para que el propio alumno llegue a la solución.

◆ **Agrupamiento:** estos alumnos participarán en todas las actividades que sea posible con sus compañeros, con el seguimiento del desarrollo de su currículo, introduciendo las modificaciones oportunas que potencien la adecuación en sus aprendizajes.

ORGANIZACIÓN Y SEGUIMIENTO DE LOS PLANES DE RECUPERACIÓN DEL ALUMNADO CON MATERIAS PENDIENTES DE CURSOS ANTERIORES

Para aquellos alumnos que no hayan superado la asignatura Programación se establecerá, al inicio de curso, un plan de recuperación personalizado. Este consistirá en la realización de ejercicios debidamente explicados que tendrán que entregarse en tiempo y forma siguiendo las instrucciones acordadas en el departamento.

LIBROS O MATERIALES VAN A SER UTILIZADOS PARA EL DESARROLLO DE LA MATERIA

Nombre	ISBN
No se utilizará	

ACTIVIDADES EXTRAESCOLARES/COMPLEMENTARIAS QUE SE VAN A LLEVAR A CABO

Nombre	Inicio	Fin
Semana Europea de la programación	06/11/2023	30/11/2023
Actividades de aprendizaje de pensamiento computacional durante la semana Europea de la programación CodeWeek		

UNIDADES DE PROGRAMACIÓN

Las unidades de programación organizan la acción didáctica orientada hacia la adquisición de competencias. En este proceso se desarrollan los saberes básicos (conocimientos, destrezas y actitudes), cuyo aprendizaje resulta necesario para la adquisición de competencias.

Los saberes básicos desarrollados en cada unidad de programación son impartidos en clase a través de las denominadas situaciones de aprendizaje. Éstas, a su vez, se evalúan a través de procedimientos de evaluación; los utilizados en esta programación didáctica son:

Según lo programado, el porcentaje de uso de los procedimientos de evaluación para obtener la calificación final del alumnado es:	
Observación sistemática:	10,00%
Pruebas de ejecución:	30,00%
Examen tradicional/Prueba objetiva/competencial:	60,00%

En este apartado, se muestran secuenciadas las diferentes unidades de programación asociadas con la materia (Programación (Optativa) de 2º Bachillerato de Ciencias y Tecnología). También se indican las fechas aproximadas de comienzo de cada una de las unidades así como el número de periodos lectivos que se estima serán necesarios para impartir la docencia correspondiente.

Comienzo aprox.	Nombre de la unidad de programación (UP)	Periodos
16-09-2025	1.- Gestión de proyectos tecnológicos.	9
09-12-2025	2.- Método computacional.	8
10-03-2026	3.- Desarrollo de programas y aplicaciones.	8

1.- GESTIÓN DE PROYECTOS TECNOLÓGICOS. (9 PERIODOS)

Esta unidad de programación está compuesta por 1 situaciones de aprendizaje que son descritas a continuación.

GESTIÓN DE PROYECTOS TECNOLÓGICOS.

Descripción y saberes básicos de la situación de aprendizaje, integrando metodologías:

- Fases de un proyecto. El proyecto de desarrollo de software.
- Estructuras organizativas de grupos de trabajo.
- Metodologías de gestión de proyectos.

Metodología:

Trabajo en clase abierto con estrategias de trabajo cooperativo y fases de ABP para llegar a profundización del conocimiento adquirido constructivamente.

Producto solicitado a los alumnos en la situación de aprendizaje:

Preparación y memoria técnica de un proyecto grupal, realizado mediante trabajo colaborativo.

Competencias específicas que se van a trabajar en esta situación de aprendizaje:

1.- Planificar, desarrollar y gestionar proyectos de índole computacional y de desarrollo de software, siguiendo los pasos de la metodología tecnológica, de manera individual o cooperando cuando sea necesario, para dar soluciones a aspectos relacionados con la sociedad en situaciones de la vida cotidiana, profesional o industrial.

2.- Plantear y definir necesidades, observando, buscando y seleccionando 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 determinar problemas tecnológicos e iniciar procesos de creación de soluciones a partir de la información obtenida.

En esta situación de aprendizaje se va a llevar a cabo (al menos) 1 actividad:

Nombre de la actividad

Para evaluar el desarrollo de la actividad se hacen uso de procedimientos de evaluación. Estos procedimientos de evaluación miden la adquisición de las competencias por parte del alumnado utilizando los denominados criterios de evaluación.

A continuación se describen los procedimientos de evaluación con sus criterios asociados:

Tipo	Nombre	Criterios evaluados (peso)
------	--------	----------------------------

Tipo	Nombre	Criterios evaluados (peso)
Observación sistemática	Proyecto tecnológico colaborativo	1.1.- Diseñar la planificación, ejecución, seguimiento y control de las fases de un proyecto, realizando una coordinación adecuada y aplicando metodologías propias de la gestión de proyectos. (1) 1.2.- Gestionar con diligencia grupos de trabajo, estableciendo la estructura organizativa conveniente, determinando responsabilidades y asignando recursos de una manera eficiente, adoptando empatía y liderazgo. (1) 1.3.- Cooperar dentro de un grupo de proyecto, asumiendo responsablemente una función concreta y valorando constructivamente aportaciones del resto de miembros del grupo. (1) 2.1.- Observar, plantear y definir problemas o necesidades sobre fenómenos sociales cotidianos y/o profesionales de los diferentes sectores productivos, que puedan ser respondidas utilizando métodos tecnológicos. (1) 2.2.- Buscar y contrastar información procedente de diferentes fuentes de manera crítica, evaluando su fiabilidad y pertinencia. (1) 2.3.- Comprender, examinar y analizar objetos y sistemas, empleando el proceso tecnológico y utilizando herramientas de simulación en la construcción de conocimiento. (1) 2.4.- Adoptar medidas preventivas, éticas y críticas para la protección de los datos, la salud y el medioambiente, identificando riesgos en los problemas planteados. (1)
Pruebas de ejecución	Trabajo colaborativo y actitud de trabajo en equipo	1.1.- Diseñar la planificación, ejecución, seguimiento y control de las fases de un proyecto, realizando una coordinación adecuada y aplicando metodologías propias de la gestión de proyectos. (3) 1.2.- Gestionar con diligencia grupos de trabajo, estableciendo la estructura organizativa conveniente, determinando responsabilidades y asignando recursos de una manera eficiente, adoptando empatía y liderazgo. (3) 1.3.- Cooperar dentro de un grupo de proyecto, asumiendo responsablemente una función concreta y valorando constructivamente aportaciones del resto de miembros del grupo. (3) 2.1.- Observar, plantear y definir problemas o necesidades sobre fenómenos sociales cotidianos y/o profesionales de los diferentes sectores productivos, que puedan ser respondidas utilizando métodos tecnológicos. (3) 2.2.- Buscar y contrastar información procedente de diferentes fuentes de manera crítica, evaluando su fiabilidad y pertinencia. (3) 2.3.- Comprender, examinar y analizar objetos y sistemas, empleando el proceso tecnológico y utilizando herramientas de simulación en la construcción de conocimiento. (3) 2.4.- Adoptar medidas preventivas, éticas y críticas para la protección de los datos, la salud y el medioambiente, identificando riesgos en los problemas planteados. (3)

Tipo	Nombre	Criterios evaluados (peso)
Examen tradicional/Prueba objetiva/competencial	Creación de rutina en bloques	1.1.- Diseñar la planificación, ejecución, seguimiento y control de las fases de un proyecto, realizando una coordinación adecuada y aplicando metodologías propias de la gestión de proyectos. (6) 1.2.- Gestionar con diligencia grupos de trabajo, estableciendo la estructura organizativa conveniente, determinando responsabilidades y asignando recursos de una manera eficiente, adoptando empatía y liderazgo. (6) 1.3.- Cooperar dentro de un grupo de proyecto, asumiendo responsablemente una función concreta y valorando constructivamente aportaciones del resto de miembros del grupo. (6) 2.1.- Observar, plantear y definir problemas o necesidades sobre fenómenos sociales cotidianos y/o profesionales de los diferentes sectores productivos, que puedan ser respondidas utilizando métodos tecnológicos. (6) 2.2.- Buscar y contrastar información procedente de diferentes fuentes de manera crítica, evaluando su fiabilidad y pertinencia. (6) 2.3.- Comprender, examinar y analizar objetos y sistemas, empleando el proceso tecnológico y utilizando herramientas de simulación en la construcción de conocimiento. (6) 2.4.- Adoptar medidas preventivas, éticas y críticas para la protección de los datos, la salud y el medioambiente, identificando riesgos en los problemas planteados. (6)

2.- MÉTODO COMPUTACIONAL. (8 PERIODOS)

Esta unidad de programación está compuesta por 1 situaciones de aprendizaje que son descritas a continuación.

MÉTODO COMPUTACIONAL.

Descripción y saberes básicos de la situación de aprendizaje, integrando metodologías:

- Organización de la información y tipos de estructuras de datos.
- Descomposición de problemas y secuenciación.
- Generalización y reconocimiento de patrones.
- La abstracción como proceso específico, mediante representaciones simplificadas de los sistemas.
- Diseño mediante diagramas de flujos y pseudocódigo. La iteración, análisis recursivo, diseño iterativo, inducción.

Metodología:

Trabajo en clase guiado en una primera fase y abierto en una segunda para llegar a profundización del conocimiento adquirido constructivamente.

Producto solicitado a los alumnos en la situación de aprendizaje:

Prácticas sobre las bases de pensamiento computacional, utilizando las estrategias para resolver distintos problemas con dispositivos didácticos programables.

Competencias específicas que se van a trabajar en esta situación de aprendizaje:

3.- Organizar y analizar la información de un modo lógico, desde el punto de vista del pensamiento computacional, formulando y abordando problemas de forma que sus soluciones puedan ser representadas como secuencias de instrucciones y algoritmos.

4.- Identificar problemas o tareas susceptibles de ser resueltos u automatizados por medios informáticos. Discernir y concretar los requisitos del programa, idear la arquitectura del mismo y llevarlo a cabo mediante el uso lenguajes de programación de alto nivel, para posteriormente comprobar que se cumple con las especificaciones que previamente se establecieron en el análisis de requisitos.

En esta situación de aprendizaje se va a llevar a cabo (al menos) 1 actividad:

Nombre de la actividad

Para evaluar el desarrollo de la actividad se hacen uso de procedimientos de evaluación. Estos procedimientos de evaluación miden la adquisición de las competencias por parte del alumnado utilizando los denominados criterios de evaluación.

A continuación se describen los procedimientos de evaluación con sus criterios asociados:

Tipo	Nombre	Criterios evaluados (peso)
------	--------	----------------------------

Tipo	Nombre	Criterios evaluados (peso)
Observación sistemática	Resolución de problema con las 4 estrategias de PC	3.1.- Comprender, analizar y descomponer problemas, reconociendo en ellos los patrones, secuencias y condiciones que permitan formular soluciones a los mismos en forma de algoritmo. (1) 3.2.- Conocer y elaborar representaciones gráficas de algoritmos mediante diagramas de flujo. Respetado las reglas de construcción, la simbología adecuada y las estructuras de control oportunas, sin perder de vista la eficacia y eficiencia en su diseño. (1) 3.3.- Traducir, interpretar y expresar los distintos pasos que va a realizar un programa, partiendo o no de diagramas de flujo, del modo más próximo posible a un lenguaje de programación, mediante pseudocódigo. (1) 4.1.- Comprender los modos de almacenamiento, transmisión, representación y codificación de la información, así como la cuantificación de la misma. (1) 4.2.- Diferenciar algoritmo de programa informático, comprender la necesidad y el porqué de la diversidad de lenguajes informáticos existentes, así como sus tipos, para poder reconocer el ámbito de aplicación y objeto de estos. (1) 4.3.- Conocer la estructura de un lenguaje de programación de alto nivel, los tipos de datos, operadores y estructuras de control de los que se sirve, para poder ser capaz de idear, traducir y desarrollar programas informáticos. (1)
Pruebas de ejecución	Comportamiento	3.1.- Comprender, analizar y descomponer problemas, reconociendo en ellos los patrones, secuencias y condiciones que permitan formular soluciones a los mismos en forma de algoritmo. (3) 3.2.- Conocer y elaborar representaciones gráficas de algoritmos mediante diagramas de flujo. Respetado las reglas de construcción, la simbología adecuada y las estructuras de control oportunas, sin perder de vista la eficacia y eficiencia en su diseño. (3) 3.3.- Traducir, interpretar y expresar los distintos pasos que va a realizar un programa, partiendo o no de diagramas de flujo, del modo más próximo posible a un lenguaje de programación, mediante pseudocódigo. (3) 4.1.- Comprender los modos de almacenamiento, transmisión, representación y codificación de la información, así como la cuantificación de la misma. (3) 4.2.- Diferenciar algoritmo de programa informático, comprender la necesidad y el porqué de la diversidad de lenguajes informáticos existentes, así como sus tipos, para poder reconocer el ámbito de aplicación y objeto de estos. (3) 4.3.- Conocer la estructura de un lenguaje de programación de alto nivel, los tipos de datos, operadores y estructuras de control de los que se sirve, para poder ser capaz de idear, traducir y desarrollar programas informáticos. (3)

Tipo	Nombre	Criterios evaluados (peso)
Examen tradicional/Prueba objetiva/competencial	Procedimiento	3.1.- Comprender, analizar y descomponer problemas, reconociendo en ellos los patrones, secuencias y condiciones que permitan formular soluciones a los mismos en forma de algoritmo. (6) 3.2.- Conocer y elaborar representaciones gráficas de algoritmos mediante diagramas de flujo. Respetado las reglas de construcción, la simbología adecuada y las estructuras de control oportunas, sin perder de vista la eficacia y eficiencia en su diseño. (6) 3.3.- Traducir, interpretar y expresar los distintos pasos que va a realizar un programa, partiendo o no de diagramas de flujo, del modo más próximo posible a un lenguaje de programación, mediante pseudocódigo. (6) 4.1.- Comprender los modos de almacenamiento, transmisión, representación y codificación de la información, así como la cuantificación de la misma. (6) 4.2.- Diferenciar algoritmo de programa informático, comprender la necesidad y el porqué de la diversidad de lenguajes informáticos existentes, así como sus tipos, para poder reconocer el ámbito de aplicación y objeto de estos. (6) 4.3.- Conocer la estructura de un lenguaje de programación de alto nivel, los tipos de datos, operadores y estructuras de control de los que se sirve, para poder ser capaz de idear, traducir y desarrollar programas informáticos. (6)

3.- DESARROLLO DE PROGRAMAS Y APLICACIONES. (8 PERIODOS)

Esta unidad de programación está compuesta por 1 situaciones de aprendizaje que son descritas a continuación.

DESARROLLO DE PROGRAMAS Y APLICACIONES.

Descripción y saberes básicos de la situación de aprendizaje, integrando metodologías:

- Representación binaria de la información: el bit, el byte. Codificación binaria. Sistemas de numeración. Representación hexadecimal. Códigos de Entrada/Salida. Medidas de la Información.
- Almacenamiento, transmisión y tratamiento básico de la información en binario: números, texto, imágenes, ficheros.
- Lenguajes de programación: Definición. Tipos de lenguajes. Lenguajes interpretados y compilados.
- Estructura de un programa informático y elementos básicos del lenguaje: Tipos básicos de datos.
- Constantes y variables. Operadores y expresiones. Comentarios. Estructuras de control.
- Condicionales.
- Iterativas. Estructuras de datos. Funciones y bibliotecas de funciones. Manipulación de archivos.
- Orientación a objetos: Clases. Objetos. Constructores. Metodologías de desarrollo de software.
- 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.
- Computación física. Robótica: Características principales de los robots: cuerpo, control y comportamiento.
- Microcontroladores, entrada/salida, sensores, actuadores, RFID.

Metodología:

Trabajo en clase guiado en una primera fase y abierto en una segunda para llegar a profundización del conocimiento adquirido constructivamente.

Producto solicitado a los alumnos en la situación de aprendizaje:

Prácticas de iniciación de programación en distintos lenguajes interpretados y compilados.

Competencias específicas que se van a trabajar en esta situación de aprendizaje:

4.- Identificar problemas o tareas susceptibles de ser resueltos u automatizados por medios informáticos. Discernir y concretar los requisitos del programa, idear la arquitectura del mismo y llevarlo a cabo mediante el uso lenguajes de programación de alto nivel, para posteriormente comprobar que se cumple con las especificaciones que previamente se establecieron en el análisis de requisitos.

5.- Comprobar, rediseñar y proponer mejoras a soluciones computacionales propias o ajenas, de forma individual o en grupo, estableciendo la evaluación y el análisis de satisfacción del resultado final en función de criterios funcionales, así como éticos, cívicos y medioambientales para depurar el funcionamiento mediante la detección de errores o la necesidad de incluir posibles mejoras al diseño.

En esta situación de aprendizaje se va a llevar a cabo (al menos) 1 actividad:

Nombre de la actividad

Creación de página web y prácticas en distintos lenguajes de alto nivel.

Para evaluar el desarrollo de la actividad se hacen uso de procedimientos de evaluación. Estos procedimientos de evaluación miden la adquisición de las competencias por parte del alumnado utilizando los denominados criterios de evaluación.

A continuación se describen los procedimientos de evaluación con sus criterios asociados:

Tipo	Nombre	Criterios evaluados (peso)
Examen tradicional/Prueba objetiva/competencial	Página web	4.1.- Comprender los modos de almacenamiento, transmisión, representación y codificación de la información, así como la cuantificación de la misma. (6) 4.2.- Diferenciar algoritmo de programa informático, comprender la necesidad y el porqué de la diversidad de lenguajes informáticos existentes, así como sus tipos, para poder reconocer el ámbito de aplicación y objeto de estos. (6) 4.3.- Conocer la estructura de un lenguaje de programación de alto nivel, los tipos de datos, operadores y estructuras de control de los que se sirve, para poder ser capaz de idear, traducir y desarrollar programas informáticos. (6) 5.1.- Adoptar protocolos efectivos de análisis y comprobación a las soluciones computacionales desarrolladas. (6) 5.2.- Rediseñar y proponer mejoras a programas propios o ajenos que contribuyan a mejorar la solución técnica, valorando aspectos medioambientales y éticos si los hubiera. (6)
Pruebas de ejecución	Proyecto programado	4.1.- Comprender los modos de almacenamiento, transmisión, representación y codificación de la información, así como la cuantificación de la misma. (3) 4.2.- Diferenciar algoritmo de programa informático, comprender la necesidad y el porqué de la diversidad de lenguajes informáticos existentes, así como sus tipos, para poder reconocer el ámbito de aplicación y objeto de estos. (3) 4.3.- Conocer la estructura de un lenguaje de programación de alto nivel, los tipos de datos, operadores y estructuras de control de los que se sirve, para poder ser capaz de idear, traducir y desarrollar programas informáticos. (3) 5.1.- Adoptar protocolos efectivos de análisis y comprobación a las soluciones computacionales desarrolladas. (3) 5.2.- Rediseñar y proponer mejoras a programas propios o ajenos que contribuyan a mejorar la solución técnica, valorando aspectos medioambientales y éticos si los hubiera. (3)
Observación sistemática	Comportamiento	4.1.- Comprender los modos de almacenamiento, transmisión, representación y codificación de la información, así como la cuantificación de la misma. (1) 4.2.- Diferenciar algoritmo de programa informático, comprender la necesidad y el porqué de la diversidad de lenguajes informáticos existentes, así como sus tipos, para poder reconocer el ámbito de aplicación y objeto de estos. (1) 4.3.- Conocer la estructura de un lenguaje de programación de alto nivel, los tipos de datos, operadores y estructuras de control de los que se sirve, para poder ser capaz de idear, traducir y desarrollar programas informáticos. (1) 5.1.- Adoptar protocolos efectivos de análisis y comprobación a las soluciones computacionales desarrolladas. (1) 5.2.- Rediseñar y proponer mejoras a programas propios o ajenos que contribuyan a mejorar la solución técnica, valorando aspectos medioambientales y éticos si los hubiera. (1)

ANEXO I - CÁLCULO DE CALIFICACIONES

LISTADO DE COMPETENCIAS ESPECÍFICAS

La superación de Programación (Optativa) implica la adquisición de una serie de competencias específicas. Cada una de estas competencias específicas contribuirá en parte a la calificación que finalmente obtendrán sus alumnos.

No obstante, es posible que su departamento considere que una competencia específica tenga más importancia que otras en la calificación final. Esta importancia la puede fijar introduciendo un "peso" a cada competencia específica; este peso se representa por un número asociado a dicha competencia. Cuanto mayor es el peso (el número asignado) mayor es la importancia de la competencia.

A través de los criterios de evaluación se valora el grado de adquisición de cada competencia específica; la media ponderada de esas valoraciones será la calificación que el alumnado obtendrá en Programación (Optativa).

Competencias específicas	Peso
Programación (Optativa)	
1.- Planificar, desarrollar y gestionar proyectos de índole computacional y de desarrollo de software, siguiendo los pasos de la metodología tecnológica, de manera individual o cooperando cuando sea necesario, para dar soluciones a aspectos relacionados con la sociedad en situaciones de la vida cotidiana, profesional o industrial.	1
2.- Plantear y definir necesidades, observando, buscando y seleccionando 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 determinar problemas tecnológicos e iniciar procesos de creación de soluciones a partir de la información obtenida.	1
3.- Organizar y analizar la información de un modo lógico, desde el punto de vista del pensamiento computacional, formulando y abordando problemas de forma que sus soluciones puedan ser representadas como secuencias de instrucciones y algoritmos.	1
4.- Identificar problemas o tareas susceptibles de ser resueltos u automatizados por medios informáticos. Discernir y concretar los requisitos del programa, idear la arquitectura del mismo y llevarlo a cabo mediante el uso lenguajes de programación de alto nivel, para posteriormente comprobar que se cumple con las especificaciones que previamente se establecieron en el análisis de requisitos.	1
5.- Comprobar, rediseñar y proponer mejoras a soluciones computacionales propias o ajenas, de forma individual o en grupo, estableciendo la evaluación y el análisis de satisfacción del resultado final en función de criterios funcionales, así como éticos, cívicos y medioambientales para depurar el funcionamiento mediante la detección de errores o la necesidad de incluir posibles mejoras al diseño.	1

La calificación de Programación (Optativa) se calculará a través de la siguiente media ponderada:

calificación Programación (Optativa) =

$$\frac{CE1 \times 1 + CE2 \times 1 + CE3 \times 1 + CE4 \times 1 + CE5 \times 1}{1 + 1 + 1 + 1 + 1}$$

En la anterior fórmula, CE1 es la calificación que un alumno obtiene en la competencia específica 1,

En la anterior fórmula, CE2 es la calificación que un alumno obtiene en la competencia específica 2,

...

CEn sería la calificación obtenida en la competencia específica "n".

PESO ASOCIADO A CADA CRITERIO DE EVALUACIÓN

Para concretar el nivel de adquisición de cada competencia específica, se utilizarán una serie de criterios de evaluación. Así pues, las competencias no son evaluadas directamente; la evaluación se hace a través los citados criterios de evaluación; que a su vez servirán de referencia para generar la calificación obtenida por el alumnado.

Cada criterio de evaluación puede tener, a su vez, un "peso" que determina su contribución ponderada a la valoración del grado de adquisición de la competencia específica.

La calificación de cada competencia específica será la media ponderada de las calificaciones que usted otorgue a cada alumno en cada criterio de evaluación.

Competencias específicas con sus criterios de evaluación asociados	Peso
1.- Planificar, desarrollar y gestionar proyectos de índole computacional y de desarrollo de software, siguiendo los pasos de la metodología tecnológica, de manera individual o cooperando cuando sea necesario, para dar soluciones a aspectos relacionados con la sociedad en situaciones de la vida cotidiana, profesional o industrial.	
1.1.- Diseñar la planificación, ejecución, seguimiento y control de las fases de un proyecto, realizando una coordinación adecuada y aplicando metodologías propias de la gestión de proyectos.	1
1.2.- Gestionar con diligencia grupos de trabajo, estableciendo la estructura organizativa conveniente, determinando responsabilidades y asignando recursos de una manera eficiente, adoptando empatía y liderazgo.	1
1.3.- Cooperar dentro de un grupo de proyecto, asumiendo responsablemente una función concreta y valorando constructivamente aportaciones del resto de miembros del grupo.	1
2.- Plantear y definir necesidades, observando, buscando y seleccionando 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 determinar problemas tecnológicos e iniciar procesos de creación de soluciones a partir de la información obtenida.	
2.1.- Observar, plantear y definir problemas o necesidades sobre fenómenos sociales cotidianos y/o profesionales de los diferentes sectores productivos, que puedan ser respondidas utilizando métodos tecnológicos.	1
2.2.- Buscar y contrastar información procedente de diferentes fuentes de manera crítica, evaluando su fiabilidad y pertinencia.	1
2.3.- Comprender, examinar y analizar objetos y sistemas, empleando el proceso tecnológico y utilizando herramientas de simulación en la construcción de conocimiento.	1
2.4.- Adoptar medidas preventivas, éticas y críticas para la protección de los datos, la salud y el medioambiente, identificando riesgos en los problemas planteados.	1
3.- Organizar y analizar la información de un modo lógico, desde el punto de vista del pensamiento computacional, formulando y abordando problemas de forma que sus soluciones puedan ser representadas como secuencias de instrucciones y algoritmos.	
3.1.- Comprender, analizar y descomponer problemas, reconociendo en ellos los patrones, secuencias y condiciones que permitan formular soluciones a los mismos en forma de algoritmo.	1
3.2.- Conocer y elaborar representaciones gráficas de algoritmos mediante diagramas de flujo. Respetado las reglas de construcción, la simbología adecuada y las estructuras de control oportunas, sin perder de vista la eficacia y eficiencia en su diseño.	1
3.3.- Traducir, interpretar y expresar los distintos pasos que va a realizar un programa, partiendo o no de diagramas de flujo, del modo más próximo posible a un lenguaje de programación, mediante pseudocódigo.	1
4.- Identificar problemas o tareas susceptibles de ser resueltos u automatizados por medios informáticos. Discernir y concretar los requisitos del programa, idear la arquitectura del mismo y llevarlo a cabo mediante el uso lenguajes de programación de alto nivel, para posteriormente comprobar que se cumple con las especificaciones que previamente se establecieron en el análisis de requisitos.	
4.1.- Comprender los modos de almacenamiento, transmisión, representación y codificación de la información, así como la cuantificación de la misma.	1
4.2.- Diferenciar algoritmo de programa informático, comprender la necesidad y el porqué de la diversidad de lenguajes informáticos existentes, así como sus tipos, para poder reconocer el ámbito de aplicación y objeto de estos.	1
4.3.- Conocer la estructura de un lenguaje de programación de alto nivel, los tipos de datos, operadores y estructuras de control de los que se sirve, para poder ser capaz de idear, traducir y desarrollar programas informáticos.	1
5.- Comprobar, rediseñar y proponer mejoras a soluciones computacionales propias o ajenas, de forma individual o en grupo, estableciendo la evaluación y el análisis de satisfacción del resultado final en función de criterios funcionales, así como éticos, cívicos y medioambientales para depurar el funcionamiento mediante la detección de errores o la necesidad de incluir posibles mejoras al diseño.	
5.1.- Adoptar protocolos efectivos de análisis y comprobación a las soluciones computacionales desarrolladas.	1
5.2.- Rediseñar y proponer mejoras a programas propios o ajenos que contribuyan a mejorar la solución técnica, valorando aspectos medioambientales y éticos si los hubiera.	1

A modo de ejemplo, la calificación de la competencia específica 5 se calculará a través de la siguiente media ponderada:

A modo de ejemplo, la calificación de la competencia específica 5 se calculará a través de la siguiente media ponderada.

calificación CE5 =

$$\frac{\text{CEV5.1} \times 1 + \text{CEV5.2} \times 1}{1 + 1}$$

En la anterior fórmula, CEV5.1 es la calificación que un alumno ha obtenido al evaluar el criterio de evaluación 5.1, en general, CEV5.n sería la calificación obtenida en el criterio de evaluación "n".