I.E.S. SANTÍSIMA TRINIDAD

Departamento de Informática

PROGRAMACION DE COMPUTADORAS

(Asignatura de libre configuración de los centros)

1º BACHILLERATO

CURSO 2017/2018

PRESENTACIÓN

La Programación de Computadoras, es considerada por la Comisión Europea la competencia del siglo XXI, una nueva forma de alfabetización, fundamental para la comprensión de la Sociedad del Conocimiento.

Aunque el software es intangible, se trata de una de las creaciones más complejas de la humanidad, y las personas que profundicen en este conocimiento estarán mejor preparadas para integrarse activamente en un mundo en continuo proceso de transformación, en el cual la computación es motor de cambio. La Programación de Computadoras y las Tecnologías de la Información y Comunicación son materias complementarias, mientras la primera enseña al alumnado a ser creador de aplicaciones informáticas, la segunda tiene como objetivo enseñar el uso productivo y creativo de las mismas. Hay que señalar, además, que aprender Ciencias de la Computación permite conceptualizar y comprender mejor los sistemas informáticos, y por tanto hacer un uso más productivo de ellos.

El valor educativo de la materia de Programación Computadoras es doble: por un lado permite que los estudiantes sean capaces de idear, planificar, diseñar y crear software como una herramienta que permite cambiar el mundo, y por otro, desarrollar una serie de capacidades cognitivas integradas en el denominado pensamiento computacional. Esta forma de pensar enseña a razonar sobre sistemas y problemas mediante un conjunto de técnicas y prácticas bien definidas que permiten su análisis, modelado y resolución.

OBJETIVOS

1. Comprender el impacto que la programación tiene en la sociedad actual, sus aspectos positivos y negativos, y su influencia en la innovación, la comunicación y el conocimiento.

2. Comprender qué es un algoritmo, cómo son implementados en forma de programa, cómo se almacenan y ejecutan sus instrucciones, y cómo diferentes tipos de datos pueden ser representados y manipulados digitalmente.

3. Producir programas informáticos plenamente funcionales utilizando las principales estructuras de un lenguaje de programación, describiendo cómo los programas implementan algoritmos y evaluando su corrección.

4. Desarrollar y depurar aplicaciones informáticas, analizando y aplicando los principios de la ingeniería del software, utilizando estructuras de control, tipos avanzados de datos y flujos de entrada y salida en entornos de desarrollo integrados.

5. Integrarse en un equipo de desarrollo de software que sea capaz de afrontar proyectos de poca envergadura, colaborando y comunicándose con sus compañeros, fomentando sus habilidades sociales mediante la búsqueda del consenso, la negociación y la resolución de conflictos.

6. Afianzar el espíritu emprendedor con actitudes de creatividad, flexibilidad, iniciativa, trabajo en equipo, confianza en uno mismo y sentido crítico.

COMPETENCIAS

La competencia digital queda definida en el marco europeo de referencia DigComp, en donde se establecen sus cinco ámbitos de desempeño: las áreas de información, comunicación, creación de contenido, seguridad y resolución de problemas.

El carácter integrado de la competencia digital **(CD)**, permite desarrollar el resto de competencias clave de una manera adecuada. De esta forma, la materia Programación de Computadoras contribuye a la competencia

lingüística **(CL)**al definir y acotar problemas sin ambigüedades a tavés de los algoritmos. La competencia matemática y las competencias básicas en ciencia y tecnología **(CMCT)** aplicando conocimientos matemáticos, científicos y tecnológicos a la resolución de problemas en medios digitales; la competencia de aprender a aprender **(CAA)** analizando información digital y ajustando los propios procesos de aprendizaje a los tiempos y a las demandas de las tareas y actividades; las competencias sociales y cívicas **(CSC)** interactuando en comunidades y redes, y comprendiendo las líneas generales que rigen el funcionamiento de la sociedad del conocimiento; el sentido de la iniciativa y espíritu emprendedor desarrollando la habilidad para transformar ideas en proyectos **(SIEP**); y la competencia en conciencia y expresiones culturales **(CEC)** desarrollando la capacidad estética y creadora.

CONTENIDOS, CRITERIOS DE EVALUACIÓN, COMPETENCIAS Y ESTÁNDARES EVALUABLES

CONTENIDOS	CRITERIOS EVALUACIÓN Y COMPETENCIAS CLAVE	ESTANDARES EVALUABLES	INSTRUMENTOS EVALUACION
U.1. Introducción a la programación. ¿Qué es un programa? Conceptos básicos.	1. Saber qué es un programa. 2. Conocer los conceptos elementales del ámbito de la programación. CMCT, CD	1. Maneja e identifica correctamente las nociones básica en programación.	Pruebas. Tareas de clase. Observación directa.
U.2. Creando un programa Las fases del proceso de la programación. Los algoritmos. Pseudo-código y diagramas de flujo.	1. Identificar y aplicar los principales pasos crear un programa. 2. Aplicar algoritmos a la resolución de los problemas más frecuentes que se presentan al trabajar con estructuras de datos. 3. Descomponer problemas complejos en otros más simples, e idear algoritmos que permiten implementar una solución computacional. CMCT, CD, SIEP, CSC, CL.	1. Desarrolla algoritmos que permitan resolver problemas aritméticos sencillos. 2. Elabora diagramas de flujo de datos de complejidad sencilla usando elementos gráficos e interrelacionándolos entre sí para dar una respuesta a problemas concretos.	Pruebas. Tareas de clase. Observación directa. Exposición debatida.
U. 3. Tipos de datos. Tipos de datos. Constantes y variables. Operadores y expresiones.	1. Identificar, elegir y operar adecuadamente los diferentes tipos de datos en el programa. 2. Utilizar adecuadamente los operadores y expresiones CMCT, CD, CL.	1. Identifica adecuadamente los diferentes tipos de datos. 2. Aplica correctamente los operadores a expresiones determinadas para conseguir los resultados esperados.	Pruebas. Tareas de clase. Observación directa.
U.4. La programación estructurada. La programación	1. Descomponer problemas complejos en otros más simples e idear algoritmos que	1. Descompone problemas de cierta complejidad en problemas más pequeños	Pruebas. Tareas de clase.

CONTENIDOS	CRITERIOS EVALUACIÓN Y COMPETENCIAS CLAVE	ESTANDARES EVALUABLES	INSTRUMENTOS EVALUACION
estructurada. Diseño top-down. Fragmentación de problemas y algoritmos. Estructuras de control: condicionales e iterativas.	permiten implementar una solución computacional. 2. Escribir programas, convenientemente estructurados y comentados, que recogen y procesan la información procedente de diferentes fuentes y generan la correspondiente salida. 3. Analizar la estructura de programas informáticos, identificando y relacionando los elementos propios del lenguaje de programación utilizado. CMCT, CD, CCL,CL.	susceptibles de ser programados como partes separadas. 2. Escribe programas que incluyan bucles de programación para solucionar problemas que implique la división del conjunto en partes más pequeñas. 3. Obtiene el resultado de seguir un pequeño programa escrito en un código determinado partiendo de determinadas condiciones.	Observación directa. Exposición debatida.
U. 5: Lenguajes de programación. Historia y evolución de los lenguajes de programación. Tipos de lenguajes. Leguaje máquina, lenguaje bajo nivel, lenguaje alto nivel. Elementos básicos de un lenguaje.	1. Conocer y comprender la sintaxis y la semántica de las construcciones de un lenguaje de programación. 2. Distinguir entre los diferentes tipos de lenguajes. CMCT, CD, CL	1. Diferencia entre sintaxis y semántica proponiendo ejemplos concretos de un lenguaje determinado. 2. Conoce los diferentes tipos de lenguje.	Pruebas. Tareas de clase. Observación directa.
U.6 Scratch como herramienta de programación Diseño gráfico de programas.	1. Realizar pequeños programas de aplicación en Scratch aportando soluciones a problemas reales. CMCT, CD.	1. Escribe programas utilizando correctamente la gramática de Scratch. 2. Realiza programas de aplicación sencillos Scratch que solucionan problemas de la vida real.	Pruebas. Tareas de clase. Observación directa. Exposición debatida.
U.7. El lenguaje de programación Python Elementos del lenguaje.	1. Conocer y comprender la sintaxis y la semántica de las construcciones básicas de Python.. 2. Realizar pequeños programas de aplicación en Python aportando a la soluciones a problemas reales. CMCT, CD, SIEP, CL.	1. Escribe programas utilizando correctamente la gramática de Phyton. 2. Realiza programas de aplicación sencillos en Python que solucione problemas de la vida real.	Pruebas. Tareas de clase. Observación directa. Exposición debatida.

SECUENCIACIÓN Y TEMPORALIZACIÓN DE CONTENIDOS.

Primer trimestre

U.1. Introducción a la programación.

¿Qué es un programa?

Conceptos básicos.

U.2. Creando un programa

Las fases del proceso de la programación. Los algoritmos. Pseudo-código y diagramas de flujo.

U. 3. Tipos de datos.

Tipos de datos. Constantes y variables.
Operadores y expresiones.

U.6 Scratch como herramienta de programación

Diseño gráfico de programas.

La unidad 6 se dará a lo largo de todos los trimestres como soporte real del resto de unidades.

Segundo trimestre

U.4. La programación estructurada.

La programación estructurada.
Diseño top-down. Fragmentación de problemas y algoritmos. Estructuras de control: condicionales e iterativas.

U. 5: Lenguajes de programación.

Historia y evolución de los lenguajes de programación. Tipos de lenguajes. Leguaje máquina, lenguaje bajo nivel, lenguaje alto nivel. Elementos básicos de un lenguaje.

U.6 Scratch como herramienta de programación

Diseño gráfico de programas.

La unidad 6 se dará a lo largo de todos los trimestres como soporte real del resto de unidades.

Trimestre 3:

U.6 Scratch como herramienta de programación

Diseño gráfico de programas.

U.7. El lenguaje de programación Python

Elementos del lenguaje.

EVALUACION.

Rúbrica para cada bloque de la aplicación.

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 postobligatoria, 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.

TEMAS TRANSVERSALES

La Programación de Computadoras tiene un ámbito de aplicación multidisciplinar, integra conocimientos de otras materias como Matemáticas, Física, etc. y permite trabajar conocimientos relativos al patrimonio de Andalucía o a los elementos transversales del currículo como objetos de las aplicaciones informáticas a desarrollar. Permite contextualizar el proceso de enseñanza-aprendizaje a contenidos de otras materias, a los elementos transversales del currículo, o a la especialización del alumnado, propia de la etapa de Bachillerato, mediante el uso de aplicaciones y herramientas informáticas.

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

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

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

METODOLOGÍA

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

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

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

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

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

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

DIDÁCTICA

Todas las unidades de la programación de contenidos tienen la misma estructura: están divididas en una serie de apartados que siempre aparecen en el mismo orden. El objetivo con el que se han diseñado estos apartados es proponer un amplio conjunto de actividades de muy diversa índole.

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. Acceso al servidor de contenidos escuela TIC 2.0

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.

LEGISLACION.

- ✔ Ley Orgánica 8/2013 de 9 de mayo, de Educación para la Mejora de la Calidad Educativa, LOMCE.
- ✔ Real Decreto 1105/2014, de 26 de diciembre, por el que se establece le currículo básico de la Educación Secundaria Obligatoria y Bachillerato.
- ✔ Decreto 110/2016, de 14 de junio, por el que se establece la ordenación y el currículo del Bachillerato en la Comunidad Autónoma de Andalucía.
- ✔ Orden de 14 de julio de 2016, por la que se desarrolla el currículo correspondiente al Bachillerato en la Comunidad Autónoma de Andalucía, se regulan determinados aspectos de la atención a la diversidad y se establece la ordenación de la evaluación del proceso de aprendizaje del alumnado.