

Título UDI: 9. Programación por bloques con Scratch

Curso y nivel: 1º Bachillerato

Asignatura/Materia: Tecnologías de la Información y Comunicación I

Profesor: Manuel Cordón Villarejo

Temporalización: 14 sesiones.

En esta unidad didáctica se realiza:

La concreción curricular de los estándares de aprendizaje, criterios de evaluación, objetivos didácticos y contenidos para la consecución y la adquisición de competencias clave.

La transposición o aplicación didáctica de dichos elementos a la práctica educativa a través de los ejercicios, actividades, tareas y proyectos, estimulando y favoreciéndola activación de procesos cognitivos de conocimiento, comprensión, aplicación, análisis, síntesis, evaluación y creación, razonamiento y aplicación (Taxonomía de Bloom). En el caso de la enseñanza de las Tecnologías de la Información y Comunicación, dicha aplicación debe ir enfocada fundamentalmente a la realización de tareas y proyectos relacionados con la vida cotidiana.

La valoración de lo aprendido, a través de los criterios de evaluación y mediante la utilización de diversos procedimientos de evaluación entre los que se encuentra la rúbrica, se valora el grado de consecución de los estándares de aprendizaje y criterios de evaluación.

INTRODUCCIÓN

¿Qué es Scratch?

Scratch es un lenguaje de programación visual y multimedia que le facilita a los alumnos crear sus propias historias interactivas, animaciones, juegos, música y arte; además, les permite compartir con otros sus creaciones en la web. A medida que los jóvenes crean y comparten proyectos realizados en Scratch, aprenden importantes ideas matemáticas y computacionales, al mismo tiempo que aprenden a pensar creativamente, a razonar sistemáticamente y a trabajar colaborativamente.

Scratch utiliza la metáfora de “piezas encajables”, al igual que lo hacen Logo, Lego/Logo o los Etoys de Squeak para animar objetos que se encuentran en la pantalla, con un uso muy sencillo e intuitivo.

Scratch se utiliza desde un “entorno de desarrollo” que muestra de un solo golpe de vista todos los elementos necesarios: escenario, objetos y elementos del lenguaje. Permite tener tantos escenarios y objetos como se desee, utilizando aquellos que ya están disponibles con la instalación estándar de la herramienta, o bien creando otros nuevos. Este es un factor motivacional más a la hora de trabajar con el alumnado desde edades tempranas.

Prácticamente todo se lleva a cabo arrastrando y soltando elementos con el ratón, y modificando con el teclado únicamente los valores numéricos y textos.

Justificación

Esta unidad didáctica está relacionada con la unidad anterior donde se introduce a los alumnos en la programación por bloques y donde han realizado un curso interactivo en Code.

Scratch permite la programación a partir de bloques de colores, muy visuales, y sencillos de entender y utilizar, pues se basa en aprender a programar como si se estuviera jugando con piezas de LEGO. De este modo, se pueden ir agrupando bloques gráficos, sin tener que recurrir a la sintaxis o a la puntuación de los convencionales lenguajes de programación. Este entorno aprovecha los avances

en diseño de interfaces para hacer que la programación sea más atractiva y accesible para todo aquel que se enfrente por primera vez a aprender a programar.

Según sus creadores, fue diseñado como medio de expresión para ayudar a niños y jóvenes a expresar sus ideas de forma creativa. Precisamente esta facilidad es su gran virtud, pues permite realizar complejos programas en unos pocos minutos y ver los resultados al instante, al tiempo que desarrollan habilidades de pensamiento lógico y las competencias de aprendizaje del siglo XXI.

Se busca que el estudiante utilice tanto estructuras de control como el conjunto de instrucciones (bloques) que ofrece el entorno de programación Scratch para elaborar procedimientos con el fin de solucionar problemas, elaborar simulaciones o comunicar información.

Para los docentes es muy importante al aprendizaje y apropiación de este tipo de herramientas, pues le permite no sólo sean consumidores de contenido, sino que además tiene la oportunidad de ser autores de los mismos e inspirar a sus estudiantes para que ellos también sean partícipes de la construcción colaborativa de productos 3.0 de una manera creativa y divertida.

CONCRECCIÓN CURRICULAR

CRITERIOS DE EVALUACIÓN

1. Aplicar algoritmos a la resolución de los problemas más frecuentes que se presentan al trabajar con estructuras de datos. CMCT, CD.
2. Analizar y resolver problemas de tratamiento de información dividiéndolos en subproblemas y definiendo algoritmos que los resuelven. CMCT, CD.
3. Analizar la estructura de programas informáticos, identificando y relacionando los elementos propios del lenguaje de programación utilizado. CMCT, CD.
4. Conocer y comprender la sintaxis y la semántica de las construcciones básicas de un lenguaje de programación. CMCT, CD.
5. Realizar pequeños programas de aplicación en un lenguaje de programación determinado aplicándolos a la solución de problemas reales. CMCT, CD, SIEP.

ESTÁNDARES DE APRENDIZAJE EVALUABLES

- 1.1 Desarrolla algoritmos que permitan resolver problemas aritméticos sencillos elaborando sus diagramas de flujo correspondientes.
- 2.1 Escribe programas que incluyan bucles de programación para solucionar problemas que implique la división del conjunto en partes más pequeñas.
- 3.1. Obtiene el resultado de seguir un pequeño programa escrito en un código determinado, partiendo de determinadas condiciones.
- 4.1 Define qué se entiende por sintaxis de un lenguaje de programación, proponiendo ejemplos concretos de un lenguaje determinado.
- 5.1 Realiza programas de aplicación sencillos en un lenguaje determinado que solucionen problemas de la vida real.

COMPETENCIAS CLAVE

Contribución de la UDI a las competencias clave:

Competencia Digital(CD): Trabajar con Scratch permite al alumnado aprender a seleccionar, crear y manejar información de diversos tipos: texto, imágenes, secuencias animadas y sonido. Al tiempo que los estudiantes adquieren experiencia trabajando con esta información, se vuelve cada vez más receptivos y críticos analizando la información que les llega del mundo que les rodea.

Competencia Matemática en Ciencia y Tecnología (CMCT): Scratch permite aprender a través de un contexto significativo basado en el proceso de diseño. Crear un proyecto con Scratch requiere pensar una idea, dividir esa idea en pasos e implementar esos pasos mediante el sistema de programación de bloques del programa. Está diseñado para poder ver el resultado de la programación en el acto, por lo que los estudiantes aprenden este proceso de manera interactiva.

Competencia de Aprender a Aprender (CAA): Aprendiendo a programar con Scratch, los alumnos van descubriendo el razonamiento crítico y el pensamiento sistemático. En sus proyectos necesitan coordinar el tiempo y las interacciones entre diferentes personajes y su habilidad para programar esto, les proporciona una experiencia directamente relacionada con la detección de problemas, la crítica constructiva, el ensayo-error, etc.. conceptos importantes dentro del pensamiento sistemático.

En menor medida, se trabajan el resto de competencias clave como son:

Competencia en Comunicación Lingüística (CCL): Una comunicación efectiva hoy en día requiere de más habilidades que simplemente leer y escribir. Con Scratch, los alumnos aprender a ser capaces de manipular e integrar diversos tipos de información para conseguir expresarse de forma creativa y persuasiva.

Competencia en Conciencia y Expresión Cultural (CEC): Scratch estimula el pensamiento creativo, una habilidad muy valorada hoy en día. Scratch estimula a los alumnos y alumnas para que busquen soluciones innovadoras a problemas inesperados que surgen durante el proceso de diseño.

Competencia Social y Ciudadana (CSC): Al ser una herramienta que permite compartir proyectos muy fácilmente, puede ser utilizada para incentivar el debate entre los jóvenes en cuestiones con importancia social, no solo en el entorno educativo sino que se puede llevar a un nivel de discusión internacional gracias a la comunidad Scratch.

Competencia Sentido de la Iniciativa y Espíritu Emprendedor (SIEP): Desarrollando la habilidad para transformar ideas en proyectos. Al trabajar en proyectos que son significativos para los alumnos, sus propias ideas les proporcionan la motivación adecuada para sobrellevar las dificultades y retos que les plantea el proceso de diseño.

OBJETIVOS

Objetivos generales:

- 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.
- 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

Objetivos didácticos:

- Identificar y reconocer los conceptos básicos relacionados Scratch y sus bloques de programación.
- Identificar los diferentes tipos de bloques que permiten programar acciones para un juego interactivo.
- Reconocer la interfaz gráfica de Scratch 3.0 y las funciones que permiten crear historias interactivas, juegos y animaciones y compartir con otros en la web.
- Identificar y utilizar adecuadamente los objetos, la paleta de bloques, la paleta de bloques y el área de programación para realizar un juego interactivo.

CONTENIDOS

Lenguajes de programación: Estructura de un programa informático y elementos básicos del lenguaje. Tipos de lenguajes. Tipos básicos de datos. Constantes y variables. Operadores y expresiones. Comentarios. Estructuras de control. Condicionales e iterativas. Estructuras de datos. Funciones y bibliotecas de funciones. Reutilización de código. Facilidades para la entrada y salida de datos de usuario. Manipulación de archivos. Programación orientada a objetos: objetos, atributos y métodos. Interfaz gráfico de usuario. Programación orientada a eventos. Metodologías de desarrollo de software: Enfoque Top-Down, fragmentación de problemas y algoritmos. Pseudocódigo y diagramas de flujo. Depuración. Entornos de desarrollo integrado. Trabajo en equipo y mejora continua.

TRANSPOSICIÓN O APLICACIÓN DIDÁCTICA

Temporalización

Sesión 1:

Los alumnos se registran en la web del proyecto Scratch <https://scratch.mit.edu/> pulsando el enlace Únete a Scratch y después se unen a su clase. Se explica que existe una versión de escritorio que se instala en el ordenador (para Windows y Linux) y que se utiliza sin conexión a Internet y se analizan los pros y los contras de cada una de ellas y se instala la versión de Linux. A continuación se explica el entorno del programa y los elementos principales: menú principal, pestaña código (bloques de código) y tipos de bloques, área de bloques, área de edición de bloques de código (scripts), escenario (fondos), objetos (personajes), propiedades, pestañas disfraces y sonidos y mochila. Se ponen a disposición de los alumnos diversos recursos variados: guía de introducción a Scratch, manuales, ejercicios resueltos, videotutoriales, etc.. a través de la plataforma Moodle del centro y búsquedas en Internet.

Sesión 2:

Los alumnos realizan unos ejercicios básicos guiados sobre el manejo de bloques, desplazamiento de un objeto (personaje), dirección de desplazamiento, sonido, escenario.

Sesión 3:

Realización de ejercicios utilizando bucles que repiten un conjunto de acciones determinadas. Se usarán los bucles repetir, mientras, condicional (si).

Sesión 4:

Explicación de los esquemas condicionales avanzados. Bucles anidados y realización de ejercicios propuestos.

Sesión 5:

Realización de ejercicios usando animaciones que permitan introducir movimiento a un objeto (sprite)

Sesión 6:

Realización de aplicaciones interactivas que requieren la participación del usuario que envía mensajes al programa mostrando éste el resultado en pantalla.

Sesión 7:

Uso de variables en un programa para almacenar un valor. Ámbito de una variable. Operadores.

Sesión 8:

Realización de un juego sencillo donde intervienen varios objetos, paso de mensajes de unos objetos a otros y cambio de escenarios.

Sesiones 9 a 14:

Realización de un proyecto integrado basado en un juego avanzado utilizando todo lo estudiado en las sesiones anteriores.

En cada sesión se realiza un breve repaso de la sesión anterior.

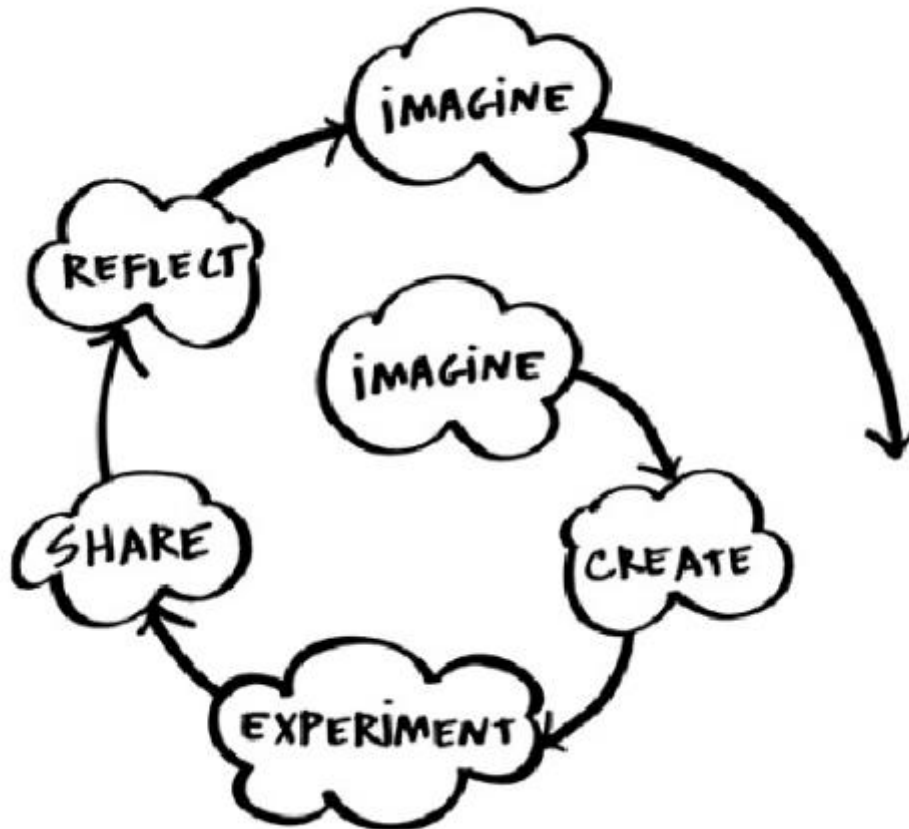
METODOLOGÍA

Trabajar con Scratch ayuda a comprender fácilmente conceptos matemáticos e informáticos que están muy bien integrados en el programa como son:

- Los procesos interactivos (bucles)
- Los criterios condicionales (si, entonces, si-no)
- Las coordenadas en un plano
- Las variables
- Etc...

Estos conceptos se aprenden dentro de un contexto significativo y motivador. Por ejemplo, no es lo mismo comprender el significado de las variables en un contexto de aprendizaje de la programación tradicional, que cuando se utilizan para el control de la visualización de una animación o en un juego que uno mismo está construyendo.

Con Scratch los alumnos y alumnas trabajan en sus propios proyectos (aprendizaje basado en proyectos ABP), de este modo aprenden sobre el proceso de diseño. Partiendo de una idea, tienen que crear el prototipo funcional, es decir, un modelo, y experimentar las soluciones que consideren necesarias para conseguir un producto final. Cuando estas ideas no funcionan tendrán que volver hacia atrás y corregir (aprendizaje basado en errores). Se crea una espiral continua: partiendo de una idea se crea un proyecto, que da lugar a nuevas ideas, que generan nuevos proyectos, y así, indefinidamente.



Estos procesos de concepción de proyectos y su construcción, desarrollan las competencias necesarias para llegar a tener:

- Un pensamiento creativo.
- Un pensamiento lógico.
- Un desarrollo de ideas, desde su concepción inicial hasta el proyecto acabado.
- Una comunicación clara.
- Un análisis sistemático.
- Capacidad de colaboración.
- Una reflexión interactiva.
- Etc..

De la misma forma que es necesario aprender no solamente a leer sino también a escribir, con la informática los alumnos y alumnas no solamente deben aprender a utilizar e interactuar con el ordenador, sino también deben aprender a crear con esta herramienta.

Lógicamente no todos los alumnos van a ser programadores profesionales, pero aprender a programar permite a los alumnos experimentar de forma creativa y ayuda al desarrollo del pensamiento lógico a la vez que comprender el funcionamiento de las nuevas tecnologías que encontrarán en su vida cotidiana.

Principios metodológicos, métodos de enseñanza y metodologías activas e innovadoras

Teniendo en cuenta la teoría genética de Piaget, la teoría social de Vigotsky, la teoría del aprendizaje significativo de Ausubel, la teoría del aprendizaje por descubrimiento de Bruner y las directrices emanadas del currículo, el proceso de enseñanza y aprendizaje lo desarrollaré:

- De forma significativa.
- Competencial.
- Trabajando tareas y proyectos integradores.
- Planteando tareas, retos, problemas y conflictos cognitivos.
- Estimulando la autonomía y el aprender a aprender.
- Empleando aprendizajes/metodologías activas, motivadoras e innovadoras como la gamificación y aula invertida (flypped classroom)
- Potenciando el aprendizaje cooperativo (trabajo colaborativo) entre los miembros del grupo profesional y la tutoría entre iguales para promover la inclusión.
- Atendiendo a la diversidad
- Con un enfoque comunicativo, potenciando la lectura, la escritura y la expresión oral.
- Integrando valores transversales de esfuerzo, respeto y coeducación, igualdad de oportunidades entre sexos, fomento de la cultura emprendedora, y haciendo referencia a las peculiaridades de Andalucía en actividades, tareas, proyectos..

A lo largo del proceso de enseñanza-aprendizaje utilizo fundamentalmente los métodos:

- Método deductivo-expositivo, para las explicaciones de contenidos.
- Método analógico, o demostrativo, para ejemplificar las prácticas.
- Método investigativo guiado, para que el alumno indague sobre las tareas/proyecto y supuestos prácticos.

Una de las cosas más interesantes de Scratch es la capacidad de utilizar materiales externos a la propia herramienta, elaborados por el alumnado, de manera que lo que ellos preparan (sus propias fotografías, sus dibujos e imágenes, su música...) puede ser incorporado al programa a través de las opciones de importación.

Otra de las características que convierten a Scratch en una herramienta colaborativa interesante es la posibilidad de compartir las creaciones que desarrollamos con la herramienta en la misma web de la aplicación, a través de la opción "Compartir", disponible desde el propio entorno de desarrollo. Para esto hace falta haber creado previamente un perfil (una cuenta de usuario) en la web oficial de Scratch. El propio programa se encarga de comprimir todo el material utilizado y enviarlo a la web para ponerlo a disposición del resto de usuarios de Scratch repartidos por el mundo.

Sin necesidad de enviar los materiales a la web oficial de Scratch, es evidente que también entre nuestro alumnado podemos fomentar la compartición de recursos, de manera que todos puedan utilizar los materiales elaborados por todos, y así crear trabajos derivados de los mismos.

Las actividades anteriores no pretenden agotar las posibilidades disponibles, sino sugerir únicamente algunas actividades que pongan en marcha al grupo. Seguro que una vez que comencemos a utilizar la herramienta, se nos ocurren muchísimas posibilidades para explorar con nuestro alumnado desde diversas áreas del currículo...

Comenzamos a poder trabajar más profundamente conceptos abstractos.

En primer lugar, podemos comenzar a asignar eventos a nuestros personajes, a utilizar el teclado y el ratón para manejar los objetos, y sobre todo a asignar acciones que ocurren cuando varios objetos interactúan entre sí. Así podremos conseguir efectos interesantes, que se irán complicando sucesivamente según se vayan adquiriendo nuevas habilidades:

Un elemento interesante que se puede trabajar son los "mensajes" que unos objetos pueden enviar a otros. Cuando se produce una condición determinada, un objeto "envía un mensaje" a otros objetos, de manera que estos lleven a cabo una acción determinada.

En todas las etapas educativas donde se vaya a trabajar cualquier herramienta que sirva para comenzar a programar un PC, hay que intentar plantear las actividades evitando que nuestro

alumnado se "lance" a programar inmediatamente con el teclado delante del PC sin hacer ningún tipo de planificación previa.

Entendemos que antes de comenzar a diseñar cualquier pequeña historia, script o aplicación, se debe comenzar haciendo un diseño básico de lo que se pretende hacer, una planificación del trabajo, una distribución del mismo, etc.

De alguna manera, este es un hábito de trabajo que se debe adquirir en general con cualquier actividad que se lleve a cabo en el aula, pero es especialmente relevante hacerlo explícito en cualquier actividad relacionada con la programación.

Esto sería importante, sobre todo en Bachillerato, donde podemos establecer un sistema de trabajo más similar al que se plantea en equipos y entornos de desarrollo reales, sin dejar por ello de mantener los factores motivacionales que hemos estado defendiendo en todas las etapas anteriores.

También se elaboran diagramas de flujo, plantillas con el pseudocódigo de la aplicación / historia que vamos a desarrollar, etc.

En definitiva, hay que establecer un sistema de trabajo en el cual:

1. Se planteen en común los objetivos de la actividad que se va a realizar.
2. Se establezca claramente el resultado final que esperamos obtener.
3. Se favorezca la creación de grupos heterogéneos en los que podamos tener alumnado diverso para favorecer también creaciones diversas.
4. Los grupos se planteen, diseñen, planifiquen y valoren qué actividad van a realizar y cuáles son los requisitos para llevarla a cabo.
5. Se establezcan fechas y rutinas de entrega de trabajos parciales, si procede.
6. Se establezcan las responsabilidades de cada miembro dentro del grupo, los momentos de trabajo individual, de trabajo en subgrupos y de trabajo en común.
7. Se lleve a cabo sesiones de planificación y se manejen "documentos" (dependiendo de la edad, éstos serán simples diseños o plantillas y cuadros más complejos...) con los requisitos, con las fechas de entrega, con los elementos diseñados, con las fórmulas de integración de los mismos, con los diagramas de flujo, los "story-boards", o lo que corresponda.
8. Se prueben los resultados obtenidos y se valoren por parte de todo el grupo, así como por agentes externos al mismo.
9. Por último, se "libere" el resultado final del proyecto que se ha llevado a cabo, y se modifique si procede o si se desea en función de las sugerencias de todos aquellos que han podido acceder al mismo.

Ejercicios, actividades, tareas, proyectos.

Los ejercicios, actividades, tareas y proyectos están graduados en dificultad y son accesibles a todo el alumnado, atendiendo así a la diversidad y al alumnado con necesidades específicas de apoyo educativo. Además, esta UDI incluye aspectos transversales e interdisciplinares.

- Los ejercicios. Son los automatismos necesarios para "saber" sobre los contenidos de la materia y son la base o el andamiaje para la posterior realización de actividades. Los alumnos realizan ejercicios sobre:
 - El manejo de los bloques.
 - El desplazamiento de personaje
 - Asignar un sonido a un objeto.
 - La dirección de desplazamiento de un objeto.
 - Bucles y sus tipos.
 - Animaciones.

- Aplicaciones interactivas.
 - Uso de variables.
- Las actividades.
Ponen en relación distintos conocimientos y activando distintos procesos cognitivos de comprensión, análisis, síntesis, inferencia, valoración, etc.. Se clasifican en:
- Motivadoras/Introducción
Introducen los conocimientos y tratan de despertar el interés en el alumnado. Algunas de las que vamos a realizar a lo largo de esta unidad son:
 - Visualización y ejecución de proyectos compartidos realizados por otros usuarios.
 - Visualización de videotutoriales sobre Scratch en Youtube.
 - Realización de ejercicios paso a paso guiados.
- Diagnósticas.
Revisión de conocimientos previos. Ejemplos:
 - Leves variaciones o modificaciones en los ejercicios guiados paso a paso.
 - Diálogos sobre la temática de la unidad didáctica y preguntas sobre diversos ejercicios de la unidad.
 - Verbalización de los procesos de pensamiento que el alumno realiza para contestar a una pregunta, resolver un problema, tomar una decisión, etc..
- De Desarrollo
Desarrolla los contenidos propuestos para la consecución de los estándares de aprendizaje, los objetivos y la adquisición de las competencias. Entre otras las actividades que se realizan en esta unidad son:
 - Dado un enunciado realiza algoritmos que resuelvan determinados problemas.
 - Diagrama de flujo de un algoritmo realizado.
- De fomento de la lectura, escritura y expresión verbal.
 - Lectura, comprensión y expresión de los conceptos abstractos relacionados con la unidad: bucle, sentencia, variable, objeto, escenario, bloque de código, disfraces, animaciones, aplicaciones interactivas, etc...
 - Lectura y comprensión de los enunciados de los ejercicios, de foros o comunidades, manuales, etc..
- De refuerzo y ampliación.
Para atender a la diversidad:

Entre las actividades de refuerzo podemos destacar:
 - Repaso de actividades de desarrollo y algoritmos básicos que resuelven un problema.
 - Participación en diálogos y debate sobre los procesos y procedimientos de resolución de problemas y actividades.
 Entre las actividades de ampliación podemos destacar:
 - Mejoras en los algoritmos que resuelven un problema.
 - Trabajos de investigación sobre determinados aspectos de la unidad didáctica.
- Tareas
Son la aplicación de los conocimientos a la práctica. Las tareas que se van a desarrollar a lo largo de la unidad son:
 - Elaboración de distintos programas que resuelvan problemas variados sobre los contenidos teóricos estudiados en la unidad.

- Realización de diagramas de flujo de un algoritmo.
- Proyecto
Consiste en investigar sobre un tema de estudio. El proyecto se compone de diversas tareas. Se realiza un proyecto que incluye todos los aspectos estudiados en la unidad sobre un juego interactivo que debe cumplir una serie de requisitos establecidos previamente por el profesor.

VALORACIÓN DE LO APRENDIDO

Para realizar la valoración del aprendizaje del alumnado hay que tener en cuenta:

-Una evaluación inicial al comienzo de la unidad con actividades diagnósticas para comprobar los conocimientos previos que tienen los alumnos sobre la unidad.

-Los estándares de aprendizaje y los criterios de evaluación son los referentes para la calificación de la UDI. Por eso, hay que determinar:

- El peso o ponderación de los estándares de aprendizaje y criterios de evaluación que se incluyen en la unidad didáctica, ajustado a lo establecido por el departamento de manera consensuada en la programación didáctica.
- Los criterios de evaluación se evalúan mediante pruebas estandarizadas o ítems de evaluación que están graduadas en dificultad para atender a la diversidad. Estos son:
 - Preguntas de conocimiento
 - Preguntas de razonamiento
 - Problemas
 - Prácticas de aplicación
 - Debate/Proyecto
 - Verbalización del proceso de razonamiento
 - Verbalización del proceso de elaboración y conclusiones de un trabajo/proyecto/tarea.
 - Informe realizado en la libreta o portfolio.
- Estas pruebas o ítems de calificación se evalúan o corrigen obteniéndose una calificación mediante los procedimientos e instrumentos de evaluación y los que vamos a utilizar fundamentalmente son:
 - La observación del trabajo realizado y del grado de esfuerzo y responsabilidad con el trabajo encomendado.
 - La corrección de los ejercicios y actividades escritas en la libreta del módulo o portfolio.
 - La corrección de la prueba escrita u oral.
 - La valoración de las tareas y proyectos.

Así para las preguntas de conocimiento, razonamiento y problemas se corrigen con pruebas o exámenes escritos.

Los debates y verbalización del proceso de elaboración y conclusiones del proyecto/tarea/trabajo con pruebas orales.

Las prácticas se corrigen con la observación de la realización de la práctica.

Y el trabajo individual o colectivo se corrige con los informes, portfolio o libreta.

- La rúbrica, en la que se describe lo que tiene que hacer el alumnado en cada criterio de evaluación para obtener una determinada calificación en la escala 1-4, 5-6, 7-8 y 9-10. Es una tabla de doble entrada, una por cada criterio de evaluación, donde se especifican unos indicadores de logro y una calificación numérica para cada uno de ellos.
Se realiza una rúbrica para cada criterio de evaluación que se estudie en la unidad o una rúbrica genérica para todos los criterios relacionados con la unidad. Como ejemplo, voy a realizar una rúbrica del criterio de evaluación 5 que está relacionado con el estándar de aprendizaje 5.1 Realiza programas de aplicación sencillos en un lenguaje determinado que solucionen problemas de la vida real. Esta rúbrica se utiliza para evaluar el proyecto que deben realizar los alumnos al finalizar la unidad.

RÚBRICA: INDICADORES DE LOGRO			
CRITERIO DE EVALUACIÓN: 5. Realizar pequeños programas de aplicación en un lenguaje de programación determinado aplicándolos a la solución de problemas reales.			
No Conseguido	Conseguido		
CALIFICACIÓN	CALIFICACIÓN	CALIFICACIÓN	CALIFICACIÓN
0-4	5-6	7-8	9-10
No ha realizado el proyecto con la herramienta Scratch o tiene muchas dificultades y no sabe utilizarla. Proyecto sin acabar y/o que no cumple la mayoría de los requisitos.	Controla la herramienta Scratch, se desenvuelve con ella, pero ha realizado un proyecto con casi todos los requisitos, pero con dificultad.	Controla la herramienta Scratch, se desenvuelve con ella y ha realizado un proyecto con casi todos los requisitos.	Controla la herramienta Scratch, se desenvuelve con ella y ha realizado un proyecto con todos los requisitos.

La calificación de la unidad será la resultante de aplicar a cada calificación el peso o porcentaje asignados a los criterios de evaluación y estándares de aprendizaje y una vez realizadas las operaciones matemáticas necesarias.

Rúbrica general para evaluar las actividades de Scratch

	No conseguido	Conseguido		
Aspectos	CALIFICACIÓN 0-4	CALIFICACIÓN 5-6	CALIFICACIÓN 7-8	CALIFICACIÓN 9-10
Proceso (10%)	<ul style="list-style-type: none"> - Realiza correctamente menos de 3 de los 5 elementos que comprende el análisis de problemas: <ol style="list-style-type: none"> 1) Formular el problema; 2) Resultados esperados; 3) Datos disponibles; 4) Restricciones; 5) procesos necesarios - No utiliza productivamente el tiempo asignado para realizar el proyecto. No cumple con el plazo de entrega. - No colabora con sus compañeros de clase. 	<ul style="list-style-type: none"> - Realiza correctamente menos de 3 de los 5 elementos que comprende el análisis de problemas: <ol style="list-style-type: none"> 1) Formular el problema; 2) Resultados esperados; 3) Datos disponibles; 4) Restricciones; 5) procesos necesarios - La mayoría del tiempo de clase lo utiliza para realizar el proyecto. Cumple con dificultad el plazo de entrega. -Colabora con sus compañeros de clase en pocas ocasiones. 	<ul style="list-style-type: none"> -Realiza correctamente 4 de los 5 elementos que comprende el análisis de problemas: <ol style="list-style-type: none"> 1) Formular el problema; 2) Resultados esperados; 3) Datos disponibles; 4) Restricciones; 5) procesos necesarios -Utiliza productivamente el tiempo asignado para realizar el proyecto. Cumple con el plazo de entrega de este. - Colabora adecuadamente con sus compañeros de clase. 	<ul style="list-style-type: none"> -Realiza correctamente los 5 elementos que componen la plantilla de análisis de problemas: <ol style="list-style-type: none"> 1) Formular el problema; 2) Resultados esperados; 3) Datos disponibles; 4) Restricciones; 5) procesos necesarios -Utiliza productivamente el tiempo asignado para realizar el proyecto. Lo culmina antes del plazo de entrega estipulado. -Colabora con sus compañeros, incluso, fuera del tiempo de clase.
Funcionamiento (10%)	<ul style="list-style-type: none"> -El programa realizado no está completo (no cumple con lo planteado por el docente en el proyecto de clase) y no funciona. 	<ul style="list-style-type: none"> -El programa realizado no está completo (no cumple con lo planteado por el docente en el proyecto de clase) y funciona parcialmente 	<ul style="list-style-type: none"> -El programa realizado no está completo (no cumple con lo planteado por el docente en el proyecto de clase), pero funciona correctamente 	<ul style="list-style-type: none"> -El programa realizado está completo (cumple con lo planteado por el docente en el proyecto de clase) y funciona correctamente
Interfaz Gráfica (10%)	<ul style="list-style-type: none"> -El programa realizado no está organizado y su diseño es básico. -La interfaz gráfica es confusa. 	<ul style="list-style-type: none"> - El programa realizado está poco organizado, tiene un solo nivel y su diseño es simple/sencillo. - La interfaz gráfica es poco 	<ul style="list-style-type: none"> -El programa realizado está organizado, tiene dos niveles y su diseño es medianamente complejo. -La interfaz gráfica es clara pero 	<ul style="list-style-type: none"> - El programa realizado está organizado, tiene varios niveles y su diseño es complejo.

	-No permite que otras personas puedan interactuar con el programa	clara y tiene escasa relación tanto con el contenido como con el diseño del programa. -Es difícil interactuar con el programa.	tiene poca relación con el contenido y con el diseño del programa. -Es fácil interactuar con el programa.	- La interfaz gráfica es clara, tiene estructura y se adapta tanto al contenido como al diseño del programa. - Es fácil interactuar con el programa.
Creatividad (10%)	- El programa realizado se basa totalmente en el diseño e ideas de otros. No se evidencia ninguna creatividad por parte del estudiante.	- El programa realizado se basa parcialmente en el diseño e ideas de otros. El aporte en creatividad por parte del estudiante es mínimo.	-El programa realizado es original y refleja la creatividad del estudiante.	- El programa realizado es muy original y evidencia un grado de creatividad excepcional por parte del estudiante.
Programación (20%)	- El programa demuestra poca comprensión de los bloques y de cómo éstos funcionan en conjunto. -Utiliza equivocadamente las estructuras de control (secuencial, condicional, iterativa). - Los hilos de programación carecen de organización. - El programa tiene varias fallas de lógica.	- El programa demuestra alguna comprensión de los bloques y cómo éstos funcionan en conjunto. - Utiliza deficientemente las estructuras de control (secuencial, condicional, iterativa). - Los hilos de programación tienen poca organización. - El programa tiene una falla de lógica.	- El programa demuestra comprensión de los bloques y de cómo estos funcionan en conjunto para alcanzar el resultado esperado. - Utiliza apropiadamente algunas estructuras de control (secuencial, condicional, iterativa). - Los hilos de programación son lógicos y están organizados. - El programa está depurado.	- El programa evidencia comprensión avanzada de bloques y procedimientos. - Utiliza apropiadamente las estructuras de control (secuencial, condicional, iterativa). - Los hilos de programación son lógicos y están bien organizados. -El programa está correctamente depurado.
Pensamiento computacional(15%)	-La elaboración del programa no evidencia características del pensamiento computacional: -Recopila datos -Analiza datos - Representa datos - Hace abstracciones - Automatiza procesos - Simula procesos - Ejecuta tareas en paralelo	-La elaboración del programa evidencia 1 característica del pensamiento computacional: -Recopila datos -Analiza datos - Representa datos - Hace abstracciones - Automatiza procesos - Simula procesos - Ejecuta tareas en paralelo	-La elaboración del programa evidencia 2 características del pensamiento computacional: -Recopila datos -Analiza datos - Representa datos - Hace abstracciones - Automatiza procesos - Simula procesos - Ejecuta tareas en paralelo	-La elaboración del programa evidencia más de 2 características del pensamiento computacional: -Recopila datos -Analiza datos - Representa datos - Hace abstracciones - Automatiza procesos - Simula procesos - Ejecuta tareas en paralelo

<p>Publicación (5%)</p>	<p>- El programa No está publicado en el sitio Web de Scratch. -No están completos los datos que identifican el programa en Scratch.</p>	<p>El programa se encuentra publicado en una cuenta cualquiera del sitio Web de Scratch.</p>	<p>El programa se encuentra publicado en una cuenta cualquiera del sitio Web de Scratch.</p>	<p>El programa se encuentra publicado en la cuenta que el estudiante tiene en el sitio Web de Scratch. -Están completos los datos que identifican el programa en Scratch:: nombre del estudiante que lo elaboró, nombre de la Institución Educativa, grado escolar del estudiante, asignatura/materia a la que corresponde el proyecto, y corta descripción del programa.</p>
<p>Contenido del área correspondiente (20%)</p>	<p>-No incluye conceptos sobre el tema del área para la que se realiza el proyecto o, los conceptos son incorrectos.</p>	<p>-Los conceptos incluidos en el programa tienen poca relación con el tema correspondiente al área para la que se realiza el proyecto.</p>	<p>-Involucra en el programa conceptos importantes sobre el tema correspondiente al área para la que se realiza el proyecto.</p>	<p>-Hace conexiones entre los conceptos del tema correspondiente al área para la que se realiza el proyecto. Demuestra comprensión profunda.</p>

Evaluación del proceso de enseñanza o autoevaluación:

Hace referencia a la reflexión que debe realizar el profesor sobre la programación de la unidad didáctica y su propia práctica docente, con el objeto de mejorarla. Esta reflexión se debe realizar sobre diversos aspectos entre los que destacamos:

- Metodología puesta en práctica
- Propuesta de ejercicios, actividades y tareas diversas y con distinto grado de dificultad para atender a la diversidad.
- Organización adecuada del aula: espacio, tiempo, agrupamiento.
- Selección adecuada de materiales y recursos.
- Utilización de recursos TIC.
- Etc..

Debido a que esta unidad está siendo impartida actualmente no se puede realizar una evaluación del proceso de enseñanza por parte del profesor sobre los elementos de la misma que se deben mejorar para próximos cursos hasta que no finalice su estudio.