

TRABAJO FINAL DE MÁSTER

MÁSTER UNIVERSITARIO EN PROFESORADO DE SECUNDARIA

SISTEMAS ELECTRÓNICOS

CALENTAMIENTO GLOBAL CON SCRATCH Y ESCUELAS EFICIENTES CON ARDUINO

ALUMNO: JOSE IGNACIO AINZUA CEMBORAIN

TUTOR: ALFREDO PINA CALAFI

FACULTAD DE CIENCIAS HUMANAS Y SOCIALES

CURSO 2013/14

FECHA: 17/06/2014

ÍNDICE:

1. RESUMEN	1
2. INTRODUCCIÓN	2
3. MARCO TEÓRICO	3
3.1 Aprendizaje basado en proyectos (ABP)	3
4. METODOLOGÍA	11
5. HERRAMIENTAS	13
5.1 Scratch	13
5.2 Arduino	27
5.3 Scratch 4 Arduino (S4A)	33
6. APRENDIZAJE BASADO EN PROYECTOS CON SCRATCH Y S4A+ARDUINO.	40
6.1 ABP 1: Videojuego sobre el cambio climático con Scratch.	41
6.2 ABP 2: Eficiencia energética en el centro educativo utilizando S4A+Arduino	61
7. BIBLIOGRAFÍA	75
8. ANEXOS	76

1. RESUMEN

El Trabajo Final de Máster (TFM) está formado por dos proyectos con metodología ABP. El primero de ellos se realiza en la asignatura de Tecnología y en coordinación con la asignatura de Ciencias Naturales, y el segundo únicamente para Tecnología.

En la primera parte del proyecto se analiza la metodología ABP utilizada y se compara con la tradicional. Posteriormente se estudian las tres herramientas utilizadas en este proyecto como son; Scratch, Scratch for Arduino y Arduino. Se eligieron estas herramientas porque permiten un aprendizaje lúdico y divertido como modo de iniciarse en el mundo de la programación y en la de las aplicaciones electrónicas.

Finalmente se detallan dos ABP que contienen como reto o pregunta guía la temática del cambio climático o calentamiento global. Las consecuencias que conlleva este gran problema podrían servir para la historia del videojuego con Scratch, ya sea el deshielo de la Antártida, talas masivas, exceso residuos, etc.

El segundo proyecto se basará en el mismo tema del calentamiento global pero con un enfoque diferente, ya que el reto consistirá en mejorar la eficiencia energética del propio centro educativo, mediante la fabricación de una aplicación con la plataforma Arduino y programada con S4A. Podría servir para concienciar al alumnado de que nosotros somos parte del problema pero también de la solución.

2. INTRODUCCIÓN

Se ha planteado este TFM con el objetivo de generar material dentro de la metodología ABP-PBL en la asignatura de Tecnología de 4º de la ESO. La razón principal por la que me he decantado por realizar este TFM es el intentar plantear una solución a dos problemas detectados durante mi experiencia en la educación. Estos dos problemas son:

- Utilización prominente del método clásico en la educación. Lo cual produce en las aulas un alumnado pasivo, donde la información es recibida de manera unilateral principalmente. Entre los factores más importantes que inciden en el alto abandono y el bajo rendimiento escolar se encuentran los métodos de enseñanza que regularmente se han seguido en las instituciones de educación superior [1].
- Se ha detectado una falta de formación en las aulas a edades tempranas en cuanto a competencias en programación informática o computación [2].

En términos generales, la enseñanza de las diferentes experiencias educativas ha consistido en seguir procedimientos rutinarios dentro del aula. El procedimiento generalmente adoptado consiste en una exposición, por parte del profesor, de los conceptos involucrados en el tema de que se trate y encarga un trabajo extraescolar con el fin de evaluar el conocimiento adquirido por el estudiante. Se supone con esto que el estudiante adquirirá dichos conocimientos y además será capaz de usarlos en su vida profesional. Esta forma pasiva de actuación de los estudiantes promueve la memorización de los conceptos involucrados, la falta de reflexión sobre los mismos y la ausencia del desarrollo de estrategias en la resolución de problemas relacionados con la vida real y la vida profesional del estudiante[3]. Como profesores debemos de promover más la participación activa del alumnado, siendo ellos/as protagonistas de su proceso de aprendizaje.

El segundo problema detectado es la falta de formación dentro del campo de la informática. En general la mayoría de los jóvenes, actualmente, utiliza la tecnología para mantenerse en contacto con sus amigos, jugar, ver vídeos, pero muy pocos crean sus propios juegos, simulaciones o animaciones. Es como si los jóvenes que participan plenamente de la cultura digital “pudieran leer, pero no escribir” [4]. Son consumidores de la cultura digital, pero no son creadores ni autores de la misma.

Además dentro de las recomendaciones de la Unión Europea en este aspecto, como el del informe (Informatics education: Europe cannot afford to miss the boat), destaca la importancia de esta competencia y según la experiencia en diferentes países de la UE realiza las siguientes recomendaciones:

- Recomendación 1: Todos los estudiantes deben beneficiarse de la educación en la cultura digital, a partir de una edad temprana y el dominio de los conceptos básicos a los 12 años, bajo los principios y prácticas de utilizarlos de manera eficaz y ética.

- Recomendación 2. Todos los estudiantes deben beneficiarse de la educación en informática como tema científico independiente.
- Recomendación 3. La formación de profesores de gran escala debe comenzarse de manera urgente..

Otro de los grandes problemas a la hora de aprender competencias en computación son que en general los estudiantes piensan que programar es difícil, de hecho, los cursos de introducción a la programación tienen una alta tasa de abandono y normalmente los alumnos no aprenden a programar bien. Muchas personas ven la programación como una actividad técnica destinada sólo a un pequeño grupo de la población [5].

De esta manera este TFM es un proyecto ABP-PBL en el que se van a utilizar tres herramientas que nos van a servir para desarrollar de una manera lúdica y sin olvidar los objetivos de aprendizaje de computación y electrónica. Estas dos herramientas son por un lado los softwares Scratch y S4A para el desarrollo en computación, y por otro lado el hardware Arduino para desarrollar el proyecto que nos ayudaran a adquirir las competencias en electrónica.

3. MARCO TEÓRICO

3.1 Aprendizaje basado en proyectos (ABP)

El Aprendizaje Basado en Proyectos es un modelo de aprendizaje en el que los estudiantes planean, implementan y evalúan proyectos que tienen aplicación en el mundo real más allá del aula de clase [6].

Barrows (1986) define al ABP como “un método de aprendizaje basado en el principio de usar problemas como punto de partida para la adquisición e integración de los nuevos conocimientos”

Sus características fundamentales son según Barrows (1986) :

- El aprendizaje está centrado en el alumno
- El aprendizaje se produce en grupos pequeños de estudiantes
- Los profesores son facilitadores o guías.
- Los problemas forman el foco de organización y estímulo para el aprendizaje
- Los problemas son un vehículo para el desarrollo de habilidades de resolución de todo tipo de problemas.
- La nueva información se adquiere a través del aprendizaje autodirigido

Con el aprendizaje por proyectos se busca, con tareas atractivas, que sean los propios alumnos los que se motivan y comprometen con su propio aprendizaje, haciéndolo a la vez más ameno y más efectivo. Igualmente este tipo de aprendizaje hace que el alumno desarrolle una visión más completa de la asignatura y que desarrolle las competencias de una manera

transversal a sus proyectos. Con este método también se fomenta que los alumnos tomen responsabilidades ayudando a su madurez y creando un visión de las tareas interdisciplinar.

Por descontado el trabajo por proyectos ayuda a una mayor atención a la diversidad debido a que son los propios alumnos los que marcan su ritmo y a que el profesor en los horarios lectivos queda más liberado para realizar apoyos in situ.[7].

Según Redalic el aprendizaje basado en proyectos se compagina perfectamente en las asignaturas en las que se hace un uso intensivo de las tecnologías de la información y la comunicación. Este estudio revela entre otras las siguientes ventajas del ABP:

- “Desarrolla las habilidades de investigación. El Proyecto mejora ostensiblemente las aptitudes de los estudiantes para la investigación.
- Incrementa las capacidades de análisis y de síntesis, especialmente cuando el Proyecto está enfocado a que los estudiantes desarrollen estas habilidades.
- Ayuda a que los estudiantes incrementen sus conocimientos y habilidades. Se plantea y emprende una tarea desafiante que requiere de un esfuerzo sostenido durante algún tiempo.
- Aprendizaje del uso de las TIC. Los estudiantes incrementan el conocimiento y habilidades en el manejo de las TIC. El proyecto puede estar enfocado a alentar a los estudiantes en la adquisición de nuevas habilidades y conocimientos sobre las tecnologías de la información.
- Compromiso en un Proyecto. Los educandos se comprometen de forma activa y adecuadamente con la realización del trabajo de proyecto, por lo que se encuentran internamente motivados”.

Teorías del aprendizaje: Zona de desarrollo próximo

El gran mérito conceptual o teórico del ABP es integrar, aunque implícitamente, dos potentes líneas de pensamiento en psicología de la educación: la perspectiva piagetiana, centrada en el cambio conceptual y los procesos psicológicos implicados en el aprendizaje y desarrollo intelectual, y la perspectiva vygotskiana, que enfatiza las relaciones sociales y el papel de la educación en la construcción psicológica[8].

Desarrollado por Lev Vygotsky, este concepto es la base de la teoría del constructivismo. La ZDP es la diferencia de aprendizaje que puede adquirir un alumno por si mismo y la que puede adquirir ayudado por un profesor o un compañero más capacitado. En palabras del propio Lev, la zona de desarrollo próximo es “ la distancia en el nivel real de desarrollo, determinado por la capacidad de resolver independientemente un problema, y el nivel de desarrollo potencial, determinado a través de la resolución de un problema bajo la guía de un adulto o en colaboración con otro compañero más capaz”[9].

Esta idea concuerda perfectamente con el aprendizaje basado en proyectos, ya que son los propios alumnos en un principio los principales responsables de la adquisición de su propio aprendizaje, pero el profesor debe ser el orientador que haga marcar las diferencias haciendo posible que el alumnado trabaje constantemente en su zona de desarrollo próximo. De esta forma se consigue que los alumnos y alumnas aprendan de una forma natural el máximo posible en base a sus capacidades.

Metodología tradicional y ABP

Básicamente, la diferencia fundamental entre el aprendizaje tradicional y el ABP está en el carácter lineal del proceso de aprendizaje que se genera en el primero y el carácter cíclico del segundo. En el aprendizaje tradicional, la identificación de necesidades de aprendizaje y la exposición de conocimientos está a cargo del profesor (tiene principio y fin en la actividad docente). En el ABP, el alumno adquiere el máximo protagonismo al identificar sus necesidades de aprendizaje y buscar el conocimiento para dar respuesta a un problema planteado, lo que a su vez genera nuevas necesidades de aprendizaje[10].

Utilización de esta metodología como sistema de aprendizaje. La metodología del Aprendizaje Basado en Problemas intenta subsanar estas deficiencias, porque está basado en un método reflexivo de aprendizaje y no en un método memorista como el descrito anteriormente.

De esta manera se intenta promover, divulgar, investigar y evaluar metodologías de aprendizaje basadas en la resolución de problemas, a fin de que los estudiantes se apropien de los conocimientos de las experiencias educativas y los usen en la resolución de problemas de la vida real.

El ABP-PBL se ha convertido en una estrategia activa de enseñanza- aprendizaje ampliamente utilizada[8]. Es decir, el abandono de la actividad de enseñanza- aprendizaje tradicional. Y por actividad de “enseñanza- aprendizaje tradicional” donde el objetivo es transmitir, de un modo unidireccional, una serie de conocimientos, habilidades, destrezas acumuladas históricamente y valoradas por una determinada sociedad o colectivo humano y transmitidas por parte de un enseñante, el experto, a un aprendiz. Un aprendizaje centrado en el estudiante y en competencias no quiere decir que el docente no deba de ejercer un importante rol en la educación, sino que enseñar y aprender se basa en comunicar, discutir, dialogar y negociar [8]. Por lo tanto, por decirlo de alguna manera “el fin del silencio en las aulas”.

El ABP surge en los años cuarenta del siglo XX en la Facultad de Medicina de la Universidad de McMaster (Canadá). El fundamento básico de la propuesta consiste en fundamentar el aprendizaje de los alumnos y las alumnas en el trabajo individual y el trabajo en grupo (cooperativo) a través del análisis de situaciones o problemas próximos a la vida real y profesional como puede ser un caso clínico, un artículo de prensa o la

construcción de un circuito eléctrico. El procedimiento se basa en facilitar unos “objetivos de aprendizaje” a lo estudiantes, es decir, aquellos conocimientos, habilidades, competencias que deberán adquirir a lo largo de las sesiones; presentar un problema; identificar las necesidades de aprendizaje (aquello que los y las estudiantes deben saber para comprender el problema en relación a los objetivos de aprendizaje); buscar la información necesaria y regresar a los objetivos de aprendizaje[11].

Cabe destacar que los “objetivos de aprendizaje” son el “contrato” que se establece entre el docente y los y las estudiantes a partir del cual se desarrollará la evaluación, es decir, comprende aquello que el estudiante debe ser capaz de demostrar al finalizar un período de aprendizaje [11].

A nivel empírico, distintos estudios han mostrado las ventajas e inconvenientes del ABP. Basado en la percepción de los estudiantes después de participar en sesiones de ABP, Esteban (2009c) ha concluido que se perciben más ventajas que inconvenientes, diferencias que son estadísticamente significativas, valorando positivamente la estrategia con una media de 7.7 sobre 10. El aspecto positivo más mencionado fue: “el ABP permite discutir con los compañeros y trabajar cooperativamente”, mientras que el negativo fue: “requiere de más tiempo”. A pesar de que los estudiantes valoraban muy positivamente el trabajo alrededor de problemas o “casos reales”, constataron que éstos estaban alejados de la práctica y los escenarios reales donde ocurren.

Aprendizaje tradicional	Aprendizaje basado en problemas
Los profesores transmiten la información a los alumnos	Los alumnos toman la responsabilidad de aprender y crear alianzas entre alumno y profesor
Los profesores organizan el contenido en exposiciones de acuerdo a su disciplina	Los profesores diseñan su curso basado en problemas abiertos
Los alumnos son vistos como receptores pasivos de información	Los profesores buscan mejorar la iniciativa de los alumnos y motivarlos. Ven a los alumnos, como sujetos que pueden aprender por cuenta propia
Las exposiciones del profesor son basadas en comunicación unidireccional	Los alumnos trabajan en equipos para resolver problemas, adquieren y aplican el conocimiento en una variedad de contextos
El aprendizaje es individual y de competencia	Los alumnos interaccionan y aprenden en un ambiente colaborativo

Tabla 1. Comparativa entre aprendizaje tradicional y ABP

Cambio de rol del profesor:

Paradigma clásico	ABP
Explica el temario	Preparo enunciados de proyectos
Decide lo que hay que explicar	Gestiona conflictos de grupos
Prepara ejercicios	Supervisa el trabajo diario
Corrige exámenes	Ofrece Feedback frecuente
Los alumnos siguen el plan del profesor	El profesor sigue los planes de los alumnos

Aprendizaje por descubrimiento/indagación

En este tipo de aprendizaje es el alumno el encargado de la obtención de su propio aprendizaje, quedando el docente con un papel de mediador. El profesor no expone la materia para que el alumno la aprenda sino que marca una meta la cual debe ser alcanzada para los alumnos y alumnas. Con esto no se quiere decir que son los alumnos los únicos responsables de su aprendizaje ya que la labor del docente como director y mediador es fundamental.

En este tipo de aprendizaje el profesor o profesora debe dirigir el aprendizaje del alumnado sin establecer unas pautas determinadas, ya que recordemos que la responsabilidad del aprendizaje recae sobre el alumno o alumna. Debido a este punto la labor del profesorado en el aprendizaje por acción y descubrimiento es muy complicada ya que debe influir pero no demasiado en el alumnado.

Este tipo de aprendizaje fomenta el aprendizaje significativo y genera hábitos investigadores en el alumnado. Siempre y cuando se lleve a cabo con eficiencia.

El aprendizaje por acción y descubrimiento fue planteado como máximo exponente por Jerome S. Bruner nacido en Nueva York en 1915. Bruner plantea tres tipos distintos de aprendizaje por descubrimiento:

- “Descubrimiento inductivo: implica la colección y reordenación de datos para llegar a una nueva categoría, concepto o generalización.
- Descubrimiento deductivo: El descubrimiento deductivo implicaría la combinación o puesta en relación de ideas generales, con el fin de llegar a enunciados específicos, como en la construcción de un silogismo.
- Descubrimiento transductivo: En el pensamiento transductivo el individuo relaciona o compara dos elementos particulares y advierte que son similares en uno o dos aspectos”.

Aprender jugando

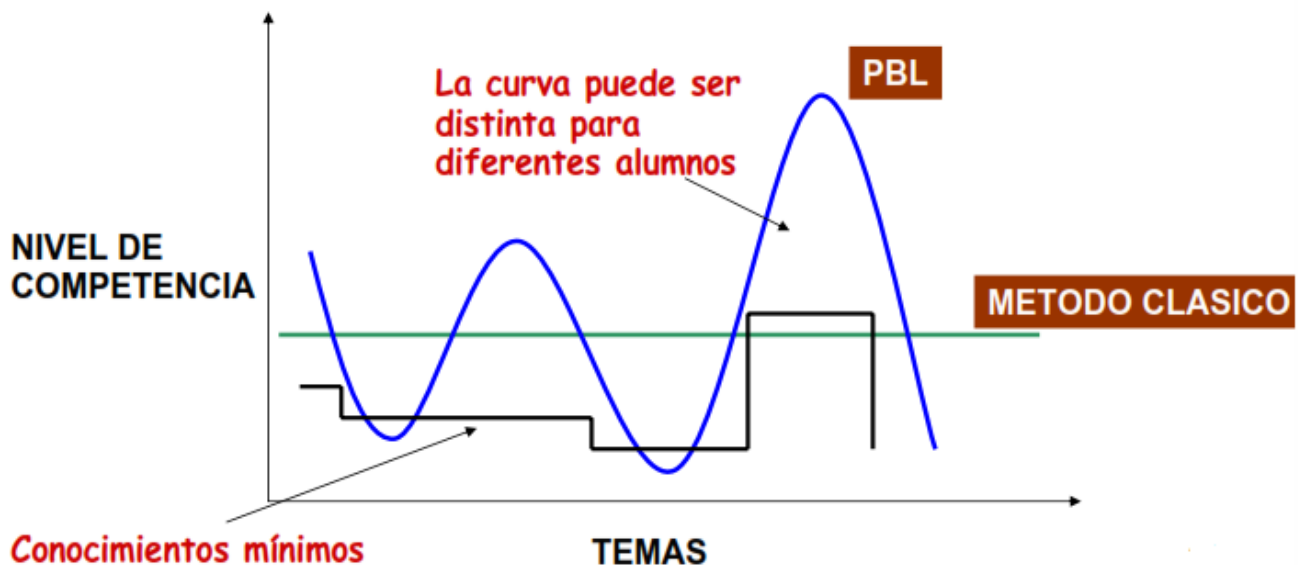
La gran ventaja de educar a través del juego es que la motivación esta intrínseca en la propia actividad, como dijo Johan Huizinga “Homo Ludens” para intentar describir de una forma más adecuada al aburrido homo sapiens como un ser humano que necesita la diversión como un ámbito más de su desarrollo tan esencial como la reflexión y el trabajo.

Considero que este tipo de aprendizaje esta muy poco desarrollado en este momento, y en base a las experiencias vividas creo que se debería potenciar en todos los entornos educativos y no solo en los de infantil y primaria.

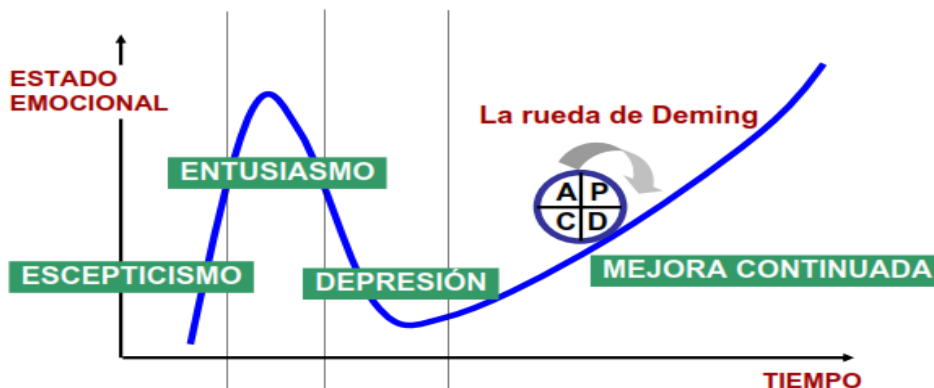
Por ello una parte transversal de los ABP que se van a plantear en este TFM tienen esta metodología ya que mediante las herramientas utilizadas como Scratch va intrínsecamente el juego inherentes a ellas.

Dificultades

- Menor cobertura de temas a cambio de mayor profundidad en otros [12]



- Un trauma para el profesor [12]



- Un trauma para el alumnado [12]:
 - El shock: “No pueden creerlo. ¿Tenemos que hacer trabajo en grupo y el profesor no va a explicar la teoría en la clase antes de que nos enfrentemos a los ejercicios?”
 - La negación. “El profesor no puede hablar en serio”.
 - La emoción fuerte: “No puedo hacerlo. Mejor abandono el curso y ya lo intentaré el próximo trimestre. O bien, No puede hacernos esto, vamos a quejarnos al director de la escuela”.
 - Resistencia y abandono: “ No voy a jugar a este juego. No me importa si me suspende”.
 - Rendición y aceptación: “Ok, me parece una estupidez pero no tengo otro remedio. Supongo que tengo que darle una oportunidad”
 - La lucha y la exploración: “Esos compañeros parece que van progresando. Quizás debo esforzarme mas o intentar cosas distintas para que me funcione tambien a mi”.
 - El retorno de la confianza: “¡Hey!, parece que estoy controlando la situación. Creo que la cosa está empezando a funcionar”.
 - Integración y éxito: “¡SI! Lo he conseguido. Ahora no entiendo por qué tenía tantas dificultades y pegas al comienzo.”
- La evaluación:
 - Nos sorprenden con sus proyectos pero nos decepcionan con sus exámenes. **Establecer catálogo de conocimientos mínimos.**
 - Algunos se nos escapan sin poder suspenderlos.(Agazapados en el seno del grupo). **Mecanismos de exigibilidad personal.**
 - Evaluación de competencias y no sólo de conocimientos.

Experiencias anteriores con ABP-PBL

La metodología PBL esta cada vez más integrada dentro del sistema educativo. Hay múltiples experiencias que sirven como respaldo de este tipo de aprendizaje.

El Aprendizaje Basado en Proyectos ha ido incorporándose a la vida sobre todo universitaria durante los últimos 50 años, especialmente en el mundo anglosajón. Las primeras experiencias se deben a la McMaster University en Notario, aplicada a los estudios de medicina. Algunas universidades del centro y el norte de Europa también han aplicado estos modelos en sus estudios: Aalborg o Roskilde en Dinamarca, Maastricht y Twente en Holanda o Linköpings en Suecia. En España también existen experiencias de este tipo especialmente en las universidades catalanas.

El ámbito de aplicación del PBL es muy variado: desde la aplicación a todo el plan de estudios, hasta la implementación en el segundo ciclo o en una asignatura o un conjunto de éstas. Por ejemplo, La Universitat Pompeu Fabra introdujo en 2004 el PBL en el primer curso de la

licenciatura en Biología, con el consiguiente rediseño de la organización horaria y de los programas de las asignaturas. La Universitat Autònoma de Barcelona, en su Escuela Universitaria de Enfermería Vall d'Hebron completó en 2004 un cambio en la metodología docente basado en el aprendizaje basado en problemas y desarrollado en grupos pequeños (8-10 estudiantes) con un tutor de acuerdo con unas fases establecidas. Este proceso se inició de forma pionera en el curso 1999 y se ha implementado de manera progresiva curso a curso. Paralelamente, se ha seguido un doble proceso de formación del profesorado y de elaboración de documentación. Este proyecto ya ha alcanzado la titulación completa. La Facultad de Medicina de la Universidad de Castilla-La Mancha ha trabajado desde su creación mediante aprendizaje basado en proyectos (PBL), asistiendo a profesores, PAS y alumnos en la formación continua específica para este tipo de enseñanza. La primera promoción salió el curso 2002-2003.

La Ingeniería de Telecomunicaciones de la Universitat Politècnica de Catalunya se estructura completamente siguiendo el modelo PBL. En concreto, en cada cuatrimestre el estudiante tiene que cursar uno o dos bloques que constituyen un bloque curricular, es decir, se evalúa globalmente. Cada bloque está formado por uno o dos módulos presenciales y un módulo no presencial denominado proyecto, que para los estudiantes supone aproximadamente la mitad de la actividad del bloque y en el cual, divididos en grupos de 5 u 8, han de analizar y resolver diferentes problemas aplicados, relacionados con los contenidos de los módulos presenciales de este bloque o de bloque anteriores.

Unida a las experiencias docentes, aparece siempre la necesidad de contar con un soporte tecnológico que facilite la implantación de estas nuevas formas de trabajo. En el mercado existe software para la gestión de proyectos y para el trabajo colaborativo, tanto libre como bajo licencia, aunque no siempre se adapta fácilmente a la utilización en las aulas o no integra la funcionalidad completa exigible.

Por ejemplo, Microsoft Project es una herramienta que permite planificar un proyecto y realizar el seguimiento del mismo.

4. METODOLOGÍA

En la metodología tiene una doble faceta en un proyecto como este: por un lado los aspectos metodológicos del modelo planteado para el proyecto que deben realizar los alumnos para superar las asignaturas, y por otro lado, la metodología aplicada para llevar a cabo esta experiencia. En cuanto a la organización del trabajo de los alumnos, planteamos la siguiente estructuración del proceso docente:

Organización en grupos

- Alumnos: Grupos de 20 alumnos máximo (dado el número actual de alumnos matriculados en las asignaturas, no será necesario más de un grupo). Equipos de 3-4 alumnos
- Profesores: Los profesores del equipo se organizan de la siguiente manera:
- *Tutores*: Cada equipo de alumnos contará con un tutor que realizará el seguimiento de su equipo.
- *Supervisores*: Los actuales profesores de las asignaturas se distribuirán las tareas de supervisión de cada una de las sesiones presenciales.
- *Expertos*: Todos los profesores del presente proyecto, a su vez, actuarán como expertos en una materia, realizando seminarios y tutorías.

Proyecto

- Un único proyecto para todos los equipos, convenientemente diseñado para que permita desarrollar toda la materia contenida en las 4 asignaturas.

Sesiones de trabajo

- Sesiones presenciales: sesiones a realizar en un aula, con los equipos de alumnos y los profesores implicados. Estas sesiones podrán ser de presentación de los problemas o actividades a desarrollar, de presentación y debate de los resultados del trabajo, o sesiones prácticas de laboratorio.
- Sesiones no presencial: Sesiones semanales, en las que cada equipo por separado, realizará de las actividades propuestas.
- Seminarios: Sesiones cortas, donde se impartirán los complementos docentes necesarios, por parte de los profesores expertos en cada tema o de expertos externos.
- Tutorías con los diferentes profesores para resolver dudas. Individuales o por equipos.

Actividades

- Resolución de los diferentes problemas o casos asociados al proyecto.
- Visitas a organismos, centros, instituciones, empresas...
- Trabajo práctico en el aula informática.
- Búsqueda de información a través de internet y en la biblioteca.
- Documentación del proyecto.

- Pruebas de evaluación.
- Cuestionarios de seguimiento.

Materiales de estudio

- Libros y manuales
- Artículos y monografías cortas
- Material informativo: prensa, webs...
- Ejercicios de autoevaluación
- Foros de debate
- Material audiovisual: videos, presentaciones...

Herramientas

- Software de desarrollo
- Software de trabajo en grupo y gestión de proyectos
- Campus Virtual
- Software ofimático: browser, procesador de textos, presentaciones, hoja de cálculo...

Evaluación

- Evaluación por el equipo docente de la carpeta del equipo, informes escritos y presentaciones orales.
- Evaluación peer review (unos equipos evalúan a otros)
- Evaluación mediante tests, planificados o por sorpresa.
- Evaluación de la actitud: Participación en el grupo y en el equipo, iniciativa, argumentación, implicación, creatividad...

Métodos

- Trabajo de los profesores en equipo.
- Determinación de expertos en cada materia.
- División de las actividades entre los profesores implicados.
- Uso del Campus Virtual, especialmente de sus utilidades para el trabajo en grupo.
- Cuestionarios y encuestas para recopilar información sobre el desarrollo de la experiencia.
- Utilización de los modelos estadísticos adecuados para el análisis de los resultados y la extracción de conclusiones.

5. HERRAMIENTAS

Para los dos ABP diseñados se van a utilizar tres herramientas, las cuales se van a describir:

- Scratch
- Scratch 4 Arduino (S4A)
- Arduino

5.1 SCRATCH

Marco teórico

Mediante esta herramienta se pretende la utilización de nuevas herramientas de aprendizaje de computación que sustituyan a los clásicos lenguajes de programación abstractos, difíciles y aburridos.

El alto nivel de abstracción y la complejidad de los conceptos que deben ser aprendidos para programar es un impedimento serio para muchas personas. Para la mayoría programar es una actividad compleja y aburrida, destinada, a lo que se denomina en lengua inglesa “computer geeks” o en nuestra lengua “frikis de la informática”.

No es de extrañar, por tanto, que algunos lenguajes de programación como Logo, y otros similares, destinados al mundo educativo no han sido utilizados de modo extensivo, ni han perdurado en el tiempo. Resnick [2] exponen las siguientes razones para que esto haya sucedido:

- Las primeras lenguas de programación eran muy difíciles de usar y muchos niños no podían aprender la sintaxis de la programación.
- La programación era introducida con actividades como, por ejemplo, generar una lista de números primos, que no resultan interesantes, ni atractivas a los jóvenes, ya que no estaban conectadas con sus intereses y experiencias.
- La programación se enseñaba en contextos donde era difícil obtener ayuda cuando algo no funcionaba.

El diseño del lenguaje de programación Scratch ha pretendido superar estas limitaciones: es un lenguaje visual y no hay que escribir líneas de programación, por tanto se evitan los errores al teclear; se pueden realizar todo tipo de proyectos y actividades personalizadas utilizando recursos multimedia; la web de Scratch permite compartir los proyectos realizados y obtener asesoramiento de otras personas.

¿Qué es Scratch?



Scratch es un programa de libre distribución, disponible en 50 lenguas diferentes, que se fundamenta en las ideas de aprendizaje constructivistas de Logo y que se puede bajar de la página Web: http://info.scratch.mit.edu/es/Scratch_1.4_Download

Scratch es un entorno de programación visual que permite a los usuarios crear **proyectos multimedia interactivos**. Un gran número de personas ha creado una amplia variedad de proyectos, utilizando Scratch, como videos musicales, presentaciones, juegos de ordenador y otro tipo de animaciones.

Los proyectos de Scratch contienen “media” y “scripts”. Las imágenes y los sonidos pueden ser importados o creados en Scratch utilizando herramientas construidas en el propio programa. La programación se realiza ensamblando bloques de comandos, de diferentes colores, para controlar objetos gráficos en 2-D llamados “duendecillos” (sprites) que se mueven en un fondo llamado “escenario” (stage). Los proyectos creados con Scratch pueden salvarse o se pueden compartir en la página Web de Scratch. (Maloney, Resnick, Rusk, Silverman & Eastmong, 2010).

Un punto fuerte de Scratch es su página Web scratch.mit.edu, se trata de una Web social, donde los usuarios que se registran pueden compartir sus proyectos, comentar proyectos realizados por otros, formar grupos con intereses comunes, agrupar proyectos en galerías y otras muchas cosas más. Esta página Web es bastante clara y explicativa, en la parte superior, encontramos los links “inicio”, “proyectos”, “galerías”, “soporte”, “foros”, “acerca de”, “mis cosas” e “idioma”.

Las lenguas de programación debían tener un “suelo bajo” (fácil empezar a programar) y un “techo alto” (oportunidades de crear proyectos más complejos con el tiempo)[13]. Además, los creadores de Scratch, Resnick dicen, que los lenguajes de programación necesitan “amplias paredes” (capacidad de englobar diferentes tipos de proyectos para que personas con intereses y estilos muy diversos se animen a programar). Para conseguir que estos objetivos sean posible, los creadores de Scratch [4] han introducido tres principios o características básicas en el diseño de este lenguaje de programación. Estos principios son: que la lengua de programación sea **lúdica**,

significativa y social. A continuación explicamos brevemente estos tres principios imbuidos en el diseño de Scratch.

El lenguaje de programación debe ser lúdico. Sus creadores comparan la forma de programar en Scratch con la forma de construir objetos con Lego. Los bloques o ladrillos de construcción de Lego tienen unos conectores que sugieren como van unidos unos con otros y es fácil jugar con ellos y empezar a construir objetos.

De modo similar, Scratch tiene unos “bloques de programación” de diferentes colores, con conectores que permiten que se puedan encajar unos con otros. El objetivo es que los niños puedan jugar con ellos desde el principio y probar a construir sencillos programas [4].

La experiencia al utilizar el lenguaje de programación debe ser significativa. Sabemos que uno de los principios del aprendizaje es que las personas aprendemos más y disfrutamos más, cuando trabajamos en proyectos personalmente significativos [4]. De este modo en el diseño de Scratch, sus creadores, han dado prioridad a dos criterios del diseño: diversidad (que pueda soportar diferentes tipos de proyectos: historias, juegos, animaciones, simulaciones) y personalización (que los proyectos se puedan personalizar importando fotos, voces, gráficos, etc.)

El uso de la lengua de programación debe propiciar la interacción social. El desarrollo de Scratch va muy unido al desarrollo de su página Web. Para que Scratch tenga éxito necesita que una gran comunidad de personas comparta, apoye, critique, colabore y pueda construir sobre el trabajo de otros. Así el concepto de “compartir” está construido en el entorno de interfaz de usuario de Scratch. Haciendo un clic sobre “compartir” los proyectos suben a la página Web de Scratch para ser compartidos.

Otras personas apoyaran, criticaran y construirán sobre los proyectos de otros, el objetivo es que finalmente resulte una experiencia de aprendizaje interactiva y enriquecedora para todos [2].

¿Por qué Scratch?

En general la mayoría de los jóvenes, actualmente, utiliza la tecnología para mantenerse en contacto con sus amigos, jugar, ver vídeos, pero muy pocos crean sus propios juegos, simulaciones o animaciones. Es como si los jóvenes que participan plenamente de la cultura digital “pudieran leer, pero no escribir”[2]. Son consumidores de la cultura digital, pero no son creadores ni autores de la misma.

Como dijimos anteriormente Scratch se basa en las ideas constructivistas de Logo. El constructivismo implica que el alumno experimente con los objetos o con herramientas que le permitan crear sus propias estrategias para aprender y resolver problemas. El aprendizaje desde el punto de vista del constructivismo es dinámico y activo, de modo que el conocimiento es construido por la persona que aprende.

Algunos conceptos claves de la teoría constructivista en la que está imbuido Scratch son el “conflicto” y el “sentido del error”. El conflicto es necesario para aprender, si no existiera conflicto no aprenderíamos. Cuando construimos cualquier conocimiento o queremos resolver un problema, pasamos por etapas de desequilibrio y reequilibrio. Para que haya desequilibrio algún hecho tiene que ocasionar un conflicto en el estudiante y ese conflicto se suele producir de forma espontánea. Justamente esto es lo que sucede cuando programamos con Scratch, de repente, algo no funciona como pensábamos. El hecho de que las cosas no funcionen como habíamos planificado origina conflicto y nos hace plantearnos estrategias de solución de problemas.

El sentido del error también es muy importante desde el punto de vista constructivista ya que significa un reto para aprender, no algo que hay que erradicar del proceso enseñanza/aprendizaje.

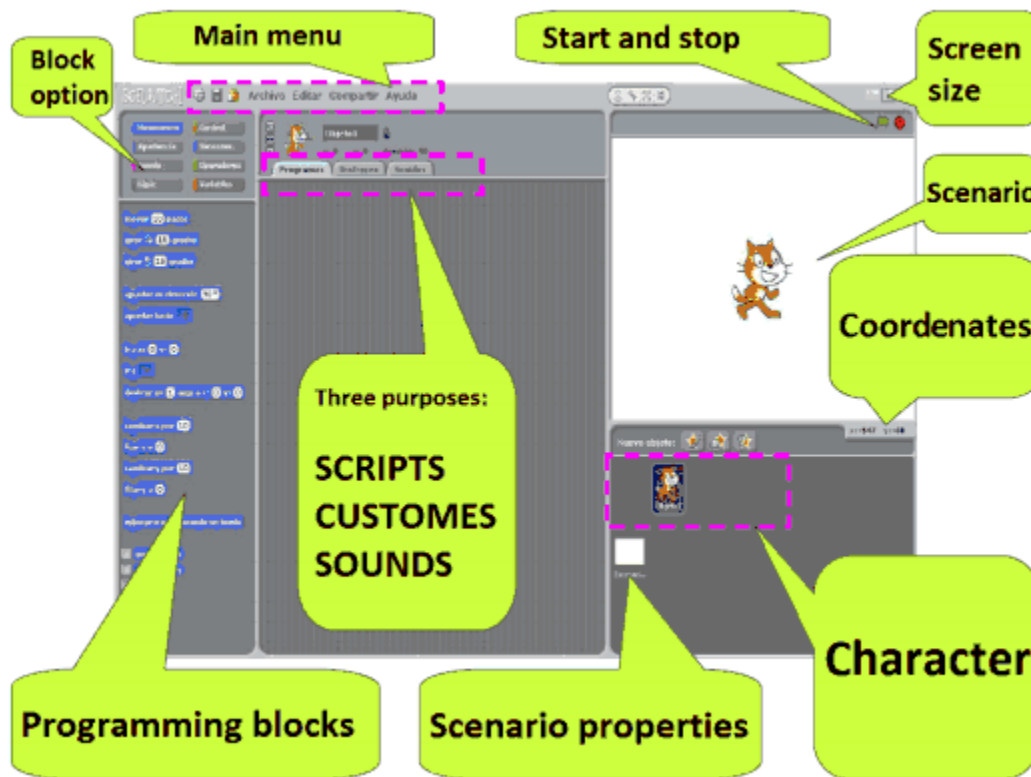
El trabajo con Scratch no se parece al trabajo del aula tradicional. Trabajar con Scratch implica actividad, comunicación e intercambio de ideas, planificación, enfrentarse a errores y plantear estrategias de solución de problemas. Ver Tabla 1.

El trabajo con Scratch	El trabajo en el aula tradicional
El estudiante es activo	El estudiante es pasivo
Comunicación e intercambio de ideas entre estudiantes	Trabajo individual con pocas posibilidades de compartir
El estudiante planifica actividades	El estudiante responde a las actividades planificadas por otros
Cada estudiante trabaja en proyectos de su interés	Los estudiantes trabajan en el proyecto asignado por el profesor
El conflicto y el error son necesarios para aprender	El conflicto y el error tienen un carácter negativo, hay que evitarlos.
Cada estudiante avanza a su propio ritmo	Todos los estudiantes deben conseguir resultados uniformes
El docente no es depositario de todo el saber. Simplemente es guía del proceso de enseñanza/aprendizaje.	El docente es el que sabe y dirige la clase.
El estudiante es cada vez más autónomo	El estudiante es totalmente dependiente

Tabla 1. Comparación de la situación de aprendizaje planteada al utilizar Scratch y la planteada en el aula tradicional [5]

En síntesis, el trabajo con Scratch ofrecerá a los estudiantes oportunidades de construir activamente sus conocimientos, planificar proyectos, plantear dudas y preguntas y trabajar en la resolución de problemas, todo ello les permitirá un aprendizaje activo y significativo.

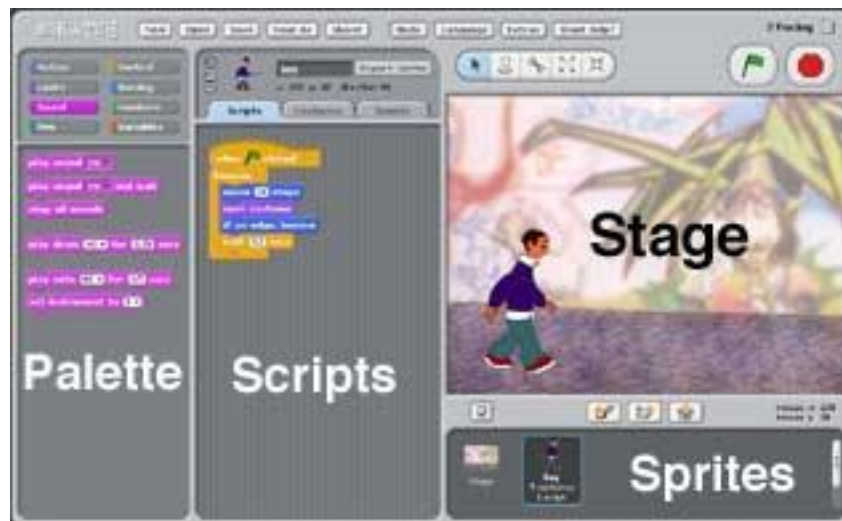
Interfaz de Scratch



El interfaz de Scratch consta de los siguientes elementos:

- **Main Menu (menú principal):** el menú principal donde encontramos (de izquierda a derecha): Cambio de Idioma, guardar, compartir proyecto, Archivo, Editar, Compartir y Ayuda.
- **Block Option (opciones de bloques):** en esta zona vemos los bloques divididos en las categorías que describiremos más adelante: Movimiento, Apariencia, Sonido, Lápiz, Control, Sensores, Operadores y Variables.
- **Programming blocks (bloques de programación):** aquí se nos presentan todos los bloques disponibles dentro de una categoría.
- **Scripts (programas):** zona donde prepararemos los programas de cada sprite.
IMPORTANTE: si hacemos doble click sobre cualquier bloque/conjunto de bloques en esta sección podemos ejecutar órdenes concretas o bloques concretos para así probar de manera más sencilla lo construido hasta el momento.
- **Costumes (disfraces):** distintas apariencias posibles para nuestro sprite.
- **Sounds (sonidos):** sonidos de los que dispone el sprite.

- Scenario properties (sprite del escenario): al seleccionarlo podremos tratarlo como a un sprite más.
- Characters (personajes/sprites): aquí están todos los personajes de nuestras historietas o juegos.
- Coordenates (coordenadas): coordenadas dentro del escenario en los ejes x e y.
- Scenario (escenario): Lugar donde podemos, de manera sencilla, situar las escenas en las que tendrán lugar nuestros videojuegos o historietas.
- Start & Stop:(bandera verde/roja) la bandera verde indica al sistema que queremos que comience a ejecutar la programación. El punto rojo detiene la ejecución en cualquier momento.
- Screen and Size: Con estos tres botones podemos modificar la estructura del interfaz, el situado a la izquierda da mayor peso a la zona de programación, el del centro lo equilibra y el de la derecha nos permite ver el escenario a pantalla completa.



Interfaz de Scratch

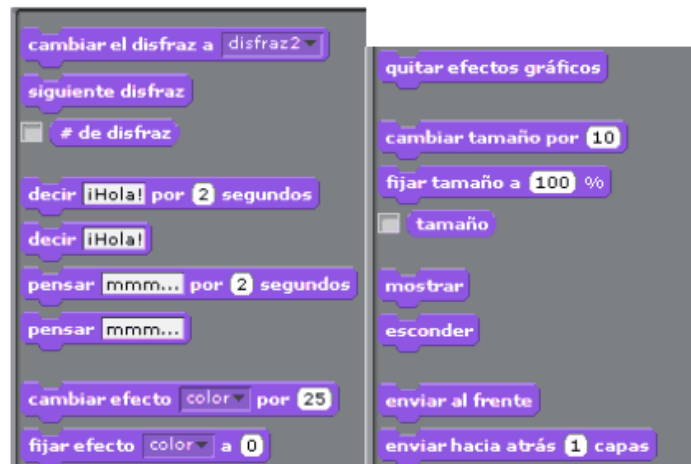
Bloques en Scratch

A continuación nombraremos e indicaremos la función de cada una de las distintas categorías que engloban los bloques anteriormente nombrados.

- Movimiento: En esta sección disponemos de todas las instrucciones que nos dan el control sobre el movimiento de los sprites.

MOVIMIENTO	
mover 10 pasos	Mueve el Objeto hacia adelante o hacia atrás.
girar ↻ 15 grados	Rota el Objeto en el sentido de las manecillas del reloj.
girar ↺ 15 grados	Rota el Objeto en el sentido contrario a las manecillas del reloj.
apuntar en dirección 90	Apunta el Objeto en la dirección especificada (0=arriba; 90=derecha; 180=abajo; -90=izquierda).
apuntar hacia	Apunta el Objeto hacia el puntero del ratón o hacia otro Objeto.
ir a x: 0 y: 0	Mueve el Objeto hacia una posición específica de X, Y en el escenario.
Ir a	Mueve el Objeto a la ubicación del puntero del ratón o de otro Objeto.
deslizar en 1 segs a x: 0 y: 0	Mueve el Objeto suavemente a una posición determinada en un lapso de tiempo específico.
cambiar x por 10	Cambia la posición X del Objeto en una cantidad determinada (incrementa).
fijar x a 0	Fija la posición X del Objeto a un valor específico.
cambiar y por 10	Modifica la posición Y del Objeto en una cantidad determinada (incrementa).
fijar y a 0	Fija la posición Y del Objeto a un valor específico.
rebotar si está tocando un borde	Gira el Objeto en sentido contrario, cuando este toca un borde del escenario.
posición x	Informa la posición X del Objeto. (Rango entre -240 a 240)
posición y	Informa la posición Y del Objeto (Rango entre -180 a 180)
dirección	Informa la dirección del Objeto (0=arriba; 90=derecha; -90=izquierda; 180=abajo)

- Apariencia: Aquí podemos modificar el aspecto de los sprites. También contiene las órdenes que corresponden a la creación de diálogos.



- Sonido: En esta zona encontraremos todo lo relacionado con el control de sonidos, en el aparecerán los sonidos propios de cada sprite.



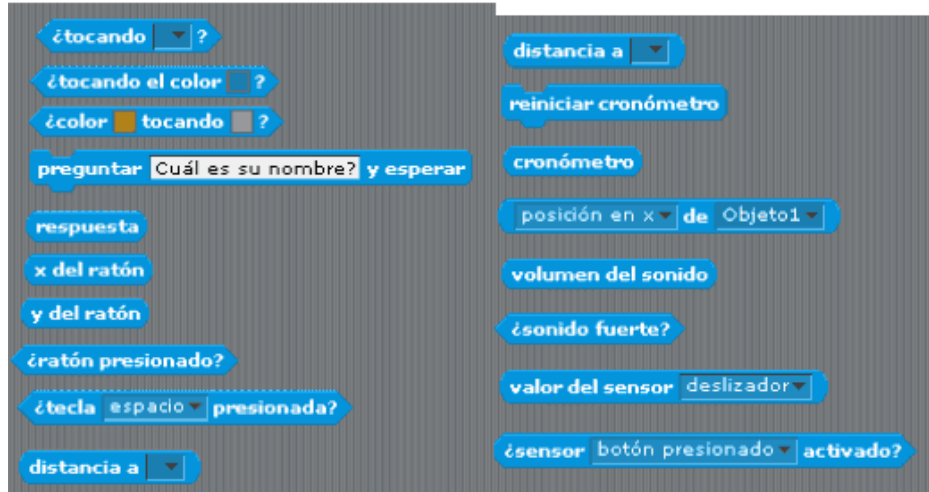
- Lápiz: Si queremos usar el escenario como un lienzo para aplicaciones de dibujo aquí encontraremos todo lo necesario



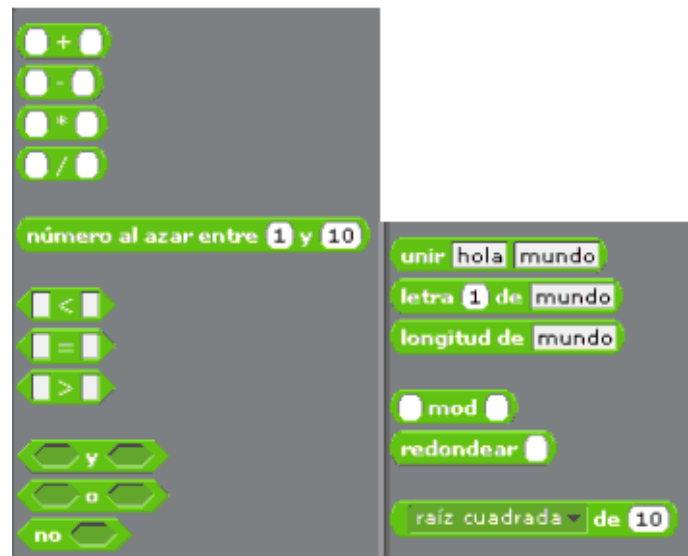
- General: Es el grupo más importante de órdenes, determinan la ejecución de nuestros programas.



- Sensores: A la hora de interactuar entre sprites o con hardware/software exterior, en este apartado tendremos lo necesario.

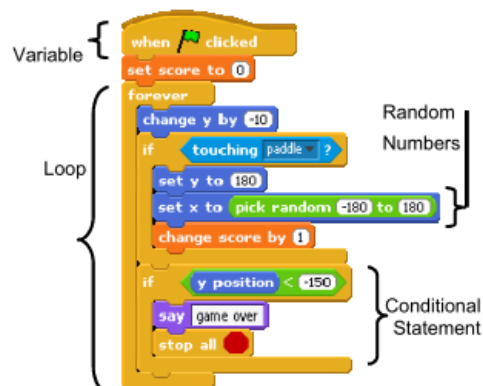


- Operadores: Agrupa todos los bloques relacionados con operaciones aritméticas y lógicas.



- Variables: En la sección de variables podemos crear variables globales o locales (de cada sprite) para los usos que necesitemos. Se explicarán detalladamente más adelante.

Primeros pasos



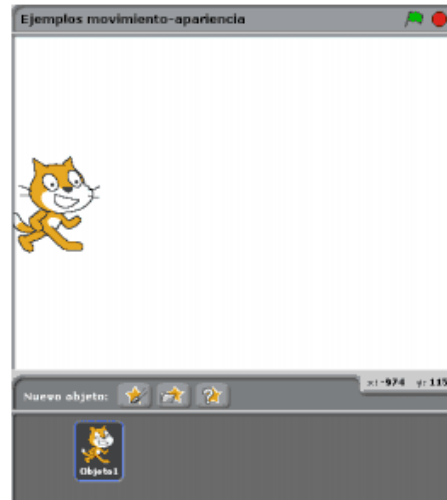
Script de Scratch para un juego de paddle

Como ya se ha mencionado anteriormente, Scratch es un lenguaje de Scripts y Sprites. Antes de nada vamos a definir estos dos conceptos.

- Cuando hablamos de un sprite, hacemos referencia a un mapa de bits dibujado en una pantalla de ordenador, es decir, nos referimos al objeto o personaje que vamos a hacer o hemos creado.
- Al hablar de un script nos referimos a la secuencia de órdenes por las que se rige el comportamiento de un sprite. Es decir, define el comportamiento de un sprite.

Para comenzar necesitamos un Sprite en el escenario. Para poner un sprite nuevo podemos hacer uso de las opciones de Nuevo Objeto que se encuentran debajo del escenario.

La primera opción nos permite dibujar usando el editor propio de Scratch, la segunda nos da la posibilidad de importar imágenes predeterminadas o personalizadas y la última opción añade uno de los predeterminados de manera aleatoria. Para este caso utilizaremos el que viene por defecto y lo situamos en el lado izquierdo del escenario, para ello lo arrastraremos desde la posición en la que aparece.



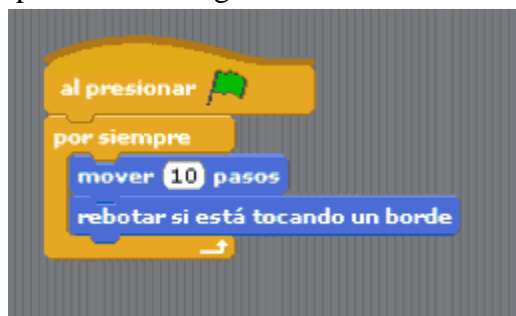
A continuación clicamos en la pestaña de programación y añadiremos el siguiente código.



El primer bloque, que pertenece al grupo general, indica al sprite que deseamos que empiece a ejecutar ese programa desde que pulsamos el botón bandera verde

(recordemos que pulsar la bandera verde significa comenzar a ejecutar el programa).

El segundo bloque indica al sprite que queremos que se desplace 10 pasos, es decir, 10 unidades en el eje x. Como podemos ver esta ejecución no aporta apenas sensación de movimiento así que vamos a probar con la siguiente.



Al código anterior le hemos añadido una nueva instrucción del grupo general que nos permite hacer las acciones que contiene de manera indefinida (o hasta que se detenga la ejecución del programa). Además le hemos añadido el bloque “rebotar si está tocando un borde” con lo cual evitamos que nuestro sprite se salga de la pantalla. Para finalizar nos interesa introducir una opción nueva del sprite, los giros.



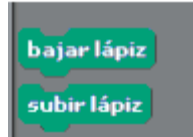
Las opciones de giros las encontramos en la parte superior del interfaz gráfico. Se trata de los tres botones de la parte izquierda. El de arriba, permite el giro del sprite en grados, el del centro lo limita sobre el eje vertical y el de más abajo no permite giros en el sprite, siempre estará orientado en la misma dirección. Esta opción es un detalle muy importante ya que en función de cual tengamos seleccionada limitará la capacidad de movimiento de nuestro sprite.

En el programa ejemplos-apariencia adjunto, se podrán encontrar una serie de scripts que nos permitirán, de manera sencilla, ver ejemplos de la amplia gama de opciones de que disponemos con tan solo el uso de los grupos de bloques: general, movimiento y apariencia.

Dibujando en el escenario

En este apartado veremos cómo convertir el escenario en un lienzo. Para ello haremos uso de los bloques en la sección de lápiz.

Los dos bloques principales de este grupo son los siguientes:

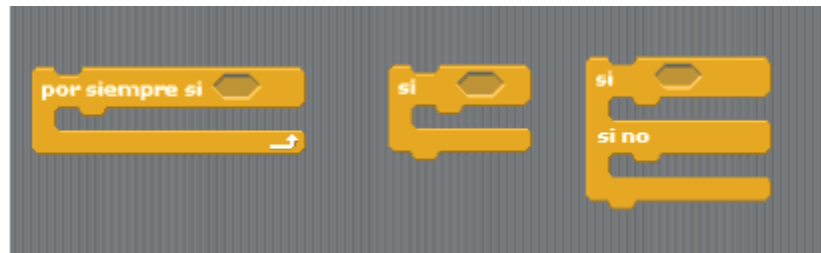


Bajar lápiz simula la acción de comenzar a escribir en el papel mientras que subir lápiz simula la acción de levantar la mano del papel. De esta manera tan sencilla simulamos la acción real de la escritura en papel. El resto de opciones nos permiten personalizar el estilo de escritura, desde el color hasta la dureza del trazo.

Se adjunta el programa Ejemplo lienzo con el que usamos el ratón como mano para dibujar. En este programa introduciremos la orden/bloque “si”.

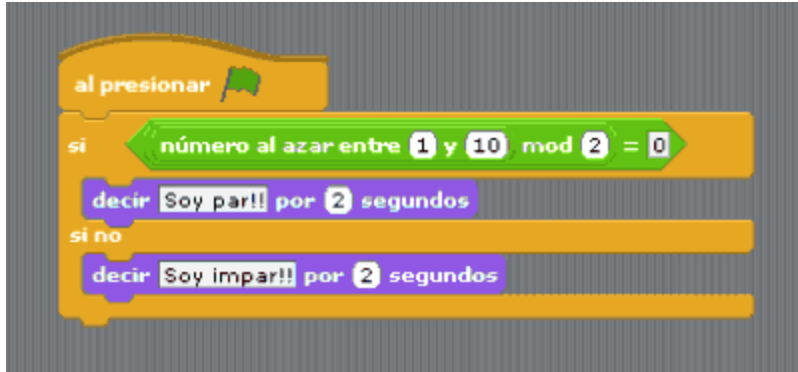


Este bloque permite sentencias y composiciones alternativas, es decir, comprueba si se cumple una condición y en función de si se cumple o no ejecutará el bloque de instrucciones que contenga. Esta instrucción la podemos encontrar de las siguientes formas:



La primera es una combinación del “por siempre” con el “si”, con ella podremos ejecutar de forma indefinida lo que contenga el bloque siempre y cuando se cumpla la condición. La segunda es la que ya conocemos y la última nos permite ejecutar dos bloques distintos de instrucciones en función de si se da la condición o no.

Como observamos de forma clara, las condiciones tienen forma “romboide”. De este modo sabemos de forma visual que tipo de instrucciones podemos usar como condiciones. Principalmente se trata de instrucciones que pertenecen al grupo de operadores y al grupo de sensores. Un ejemplo de uso de las instrucciones “si”



El siguiente script genera un aleatorio entre 1 y 10, lo divide de forma entera entre 2 y en función de si el resultado es 0 o no indica si es par o impar.

Por último encontraremos una instrucción de color azul claro que corresponde al grupo de sensores, lo explicaremos en el siguiente apartado.

Interacción con Scratch

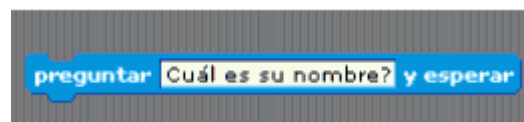
En este apartado vamos a conocer como simular interacciones en Scratch. Para ello usaremos principalmente los bloques pertenecientes al grupo sensores, pero también veremos que el grupo general contiene opciones interesantes.

Si vamos al bloque de sensores encontramos bloques tan intuitivos como los siguientes:



Con el primero podemos elegir con que sprite del programa está interactuando y con el segundo podemos hacer la misma comprobación usando colores. A continuación vamos a ver los distintos tipos de interacción:

- Entre sprites: Estos son los de la imagen anterior, nos permiten preguntar a los sprites su estado y si están interactuando con algo de su entorno. Como ya mencionamos en la sección anterior, estos bloques se suelen usar en alternativas o condicionales.
- Por teclado: Esto lo hacemos mediante el siguiente bloque:
 Esta instrucción nos permite mediante un cuadro de diálogo obtener un parámetro a través del teclado. En este caso el nombre del usuario. Este dato será guardado en el bloque respuesta que encontramos en sensores. A partir de ahora nos referiremos a ese tipo de bloques como variables, las cuales, se introducen en el siguiente apartado.



- Con el ratón: Podemos comprobar en todo momento la situación del ratón, la distancia entre el ratón y el sprite en el que se está ejecutando el código y comprobar si el botón izquierdo está siendo pulsado.

En el programa sensores y variables que se adjunta encontramos una serie de ejemplos sencillos que muestran el funcionamiento de algunos de los bloques de sensores. También se introducen conceptos de variables las cuales explicaremos a continuación.

¿ Qué son las variables?

Las variables, como en todo lenguaje de programación, son estructuras que pueden ir modificando su contenido a lo largo de la ejecución según se precise. En Scratch tenemos una serie de variables ya definidas por el propio sistema, algunos ejemplos son:



Todas las variables definidas por el sistema contiene datos a tiempo real, las 2 primeras, por ejemplo, contienen en todo momento las coordenadas del ratón en el programa.

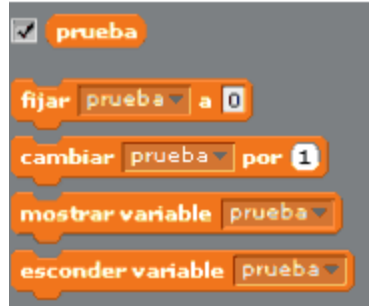
Este tipo de estructura es extremadamente útil cuando tratamos la interacción en Scratch, no solo nos permiten conocer el estado de lo que representan, sino que además nos permiten llevar cuenta de vidas, puntuación, tiempo, opciones, etc. dentro de nuestro programa.

Para trabajar con variables vamos a su grupo en el interfaz y encontraremos las opciones:



Con el primer botón crearemos una variable única con un nombre a nuestra elección y con una lista definiremos una estructura que es capaz de contener todas las variables que queramos para así poder tratarlas en grupo en vez de una en una.

Al crear una variable aparecerán las siguientes opciones:



El tick en el lado izquierdo del nombre de nuestra variable indica que, en el escenario, se está mostrando su contenido. Podemos controlar su aparición quitándole el tick o bien usando las instrucciones mostrar/esconder variable. Por otro lado, podemos fijar y cambiar el contenido de la variable con dos bloques que nos quedan, son muy útiles para resetear el valor de la variable y/o cambiar su valor según necesitemos.

En el ejemplo sensores y variables encontraremos un pequeño contador que nos permitirá entender mejor el uso de estas estructuras.

Con las listas se trabaja de forma similar añadiendo las funciones de añadir, eliminar y sustituir objetos/variables de la lista.

Con esta exposición termina la introducción al lenguaje Scratch.

5.2 ARDUINO

¿Qué es Arduino?



Arduino es una plataforma de hardware libre basada en una placa con un microcontrolador y un entorno de desarrollo principalmente utilizado para crear prototipos flexibles y de fácil uso. Fue creado para ser una plataforma de fácil uso para todo aquel interesado en crear entornos y objetos interactivos.



Figura: La imagen superior muestra la placa Arduino Uno

El hardware Arduino consiste en una placa con un microcontrolador Atmel AVR y puertos de entrada/salida. Esto lo dota de una gran sencillez de utilización y de un bajo coste.

El software de desarrollo para este dispositivo es un entorno que implementa el lenguaje de programación Processing/Wiring. Arduino puede usarse para desarrollar objetos autónomos y ser conectado con software en un ordenador (Flash por ejemplo). Además se puede programar en otros entornos como Java, ActionScript, C++, etc.

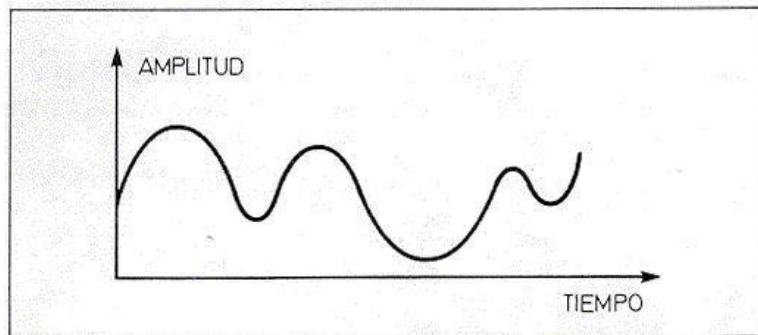
Dado que se ha trabajado sobre la placa Arduino Uno nombraremos las especificaciones de esta. Pero en cualquier caso la mayoría tienen unas características similares.

Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (Recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O pins	14 (of which 6 provide PWM output)
Analog Inputs Pins	6
DC Current per I/O pin	40mA
DC Current per 3,3V pin	50mA
Flash Memory	32 KB (ATmega328) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)

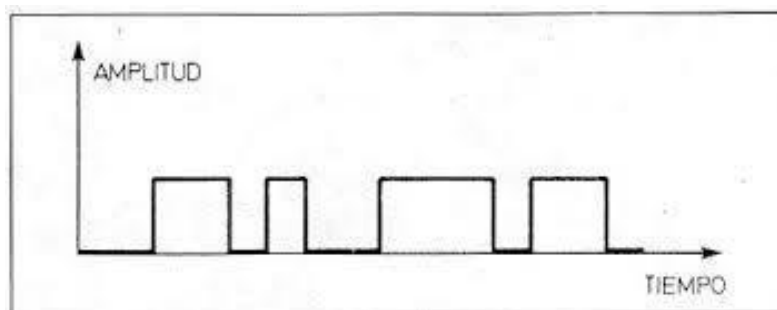
Clock Speed	16 MHz
-------------	--------

Lo más destacable de estas características, y lo que vamos a usar principalmente son las entradas y salidas (INPUTS/OUTPUTS) de señales digitales y analógicas.

Una entrada o input es un módulo que permite el envío de señales al sistema, en este caso el microcontrolador. Una salida (u output) por el contrario es el módulo encargado de mandar ese tipo de señales. Además tenemos los dos tipos de señales: digital y analógica. Una señal analógica es aquella que viene definida por una función matemática continua. Es decir, dentro del rango de una señal puede tomar todos los valores del espectro.



Mientras que la señal digital es aquella que representa valores discretos y suelen representarse mediante el uso de dos niveles, High y Low (alto y bajo o 1 y 0).



Instalación del entorno Arduino

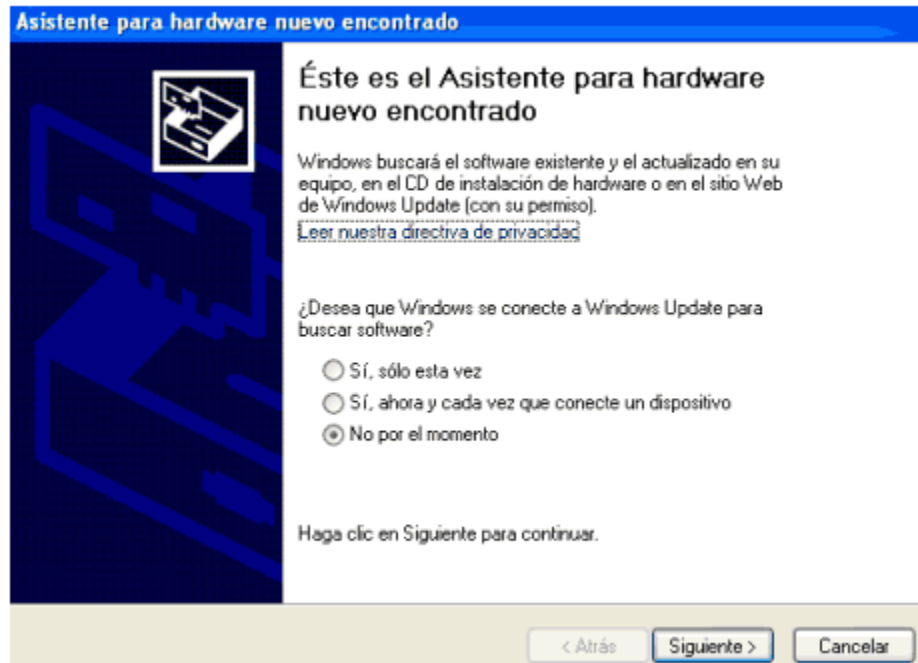
Para poder trabajar con Arduino debemos tener instalado un intérprete de Java y hacernos con la última versión del IDE de Arduino, que es de distribución gratuita.

A continuación se describe el proceso de instalación de los drivers de la placa.

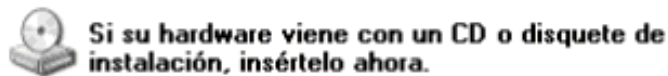
1. Conectamos la placa a un puerto USB y con ello debería encenderse un LED verde en el que estará escrito ON o PWR.
2. Una vez está bien conectado dependiendo del sistema operativo procederemos de la siguiente manera
 - a. Windows 7: El driver debería instalarse de manera automática, en caso de no ser así seguir los mismos pasos que para la instalación en Windows XP.

b. Windows XP: Al conectar la placa aparecerá el asistente para instalación de hardware de Windows.

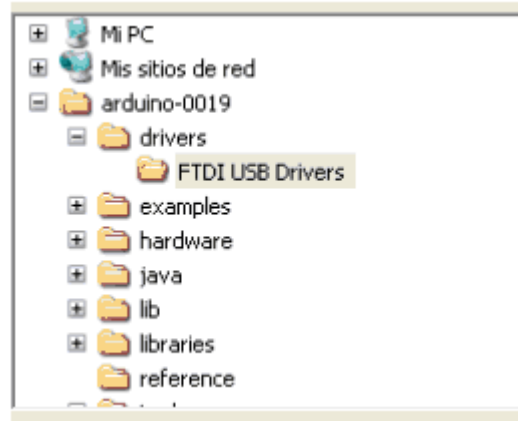
i. Elegiremos la tercera opción: “No por el momento” y pulsaremos siguiente.



ii. En la siguiente pantalla elegiremos la segunda opción: “Instalar desde una lista o ubicación específica (avanzado)”.

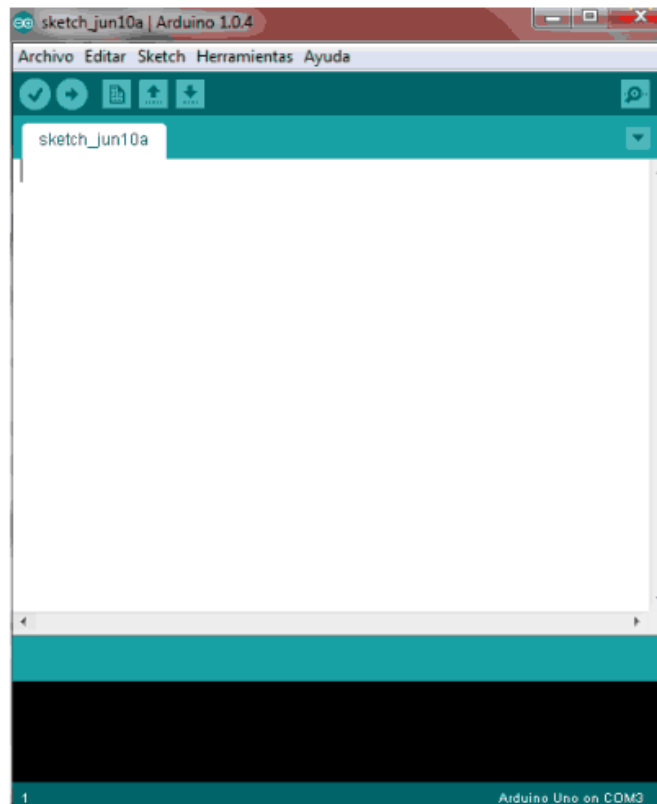


iii. A continuación nos aparecerá una pantalla donde debemos elegir la ruta del archivo/carpeta que contiene el driver que deseamos instalar. Seleccionaremos la opción “Incluir esta ubicación en la búsqueda” y buscaremos la carpeta drivers/FTDI USB Drivers



iv. Le daremos a siguiente y se instalarán los componentes necesarios para su correcto funcionamiento.

Una vez instalado correctamente procederemos a lanzar la aplicación y aparecerá el siguiente interfaz:



Este interfaz podemos dividirlo en 4 zonas:

1. Parte superior: en ella encontramos las opciones de File, Edit, Sketch, Tools y Help. Todas ellas contienen las opciones típicas de estos apartados.

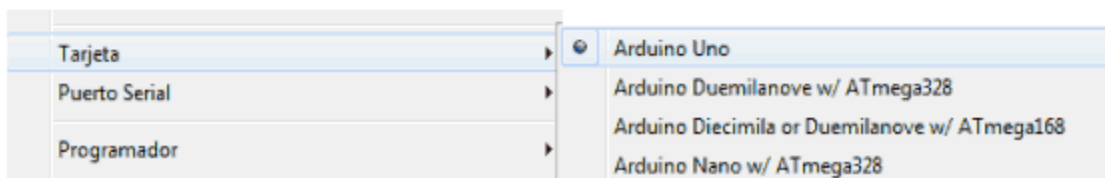
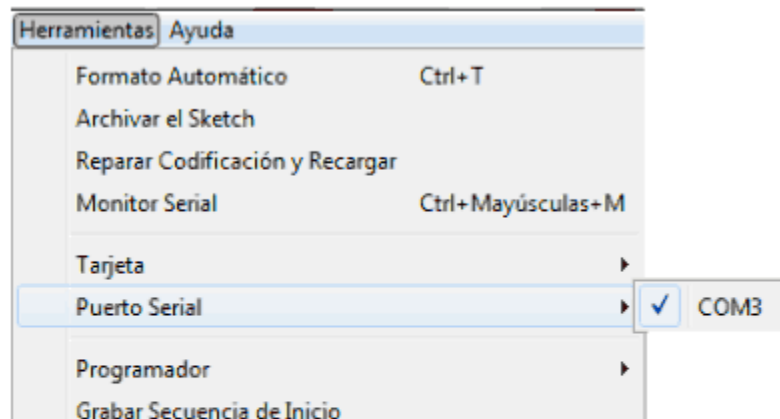
a. Archivo: nos permite trabajar con todo lo relacionado con ficheros.

También encontraremos una opción que da acceso directo a ejemplos de programación.

b. Editar: herramientas de edición de texto.

c. Sketch: es el nombre utilizado para los programas que generamos. Aquí tendremos las opciones de comprobar si es correcto y compilación.

d. Herramientas: herramientas del programa, nos permite elegir el puerto COM que queremos usar así como definir el tipo de placa que estamos usando.



e. Ayuda: colecciones de ayuda.

2. Barra de accesos directos: se sitúa debajo de la barra de herramientas. Las opciones que nos ofrece de izquierda a derecha son: Verificar, Cargar en placa, nuevo, abrir fichero, guardar fichero y, a la derecha del todo, monitorear el puerto COM.

3. Parte central: aquí escribiremos nuestros programas o Sketch.

4. Parte inferior: la ventana de color negro es la ventana de salida del código en ejecución. Nos mostrará errores de compilación o tiempos de carga en caso de que todo haya ido bien.

Lenguaje Processing

Arduino está basado en el lenguaje de programación Processing.

Processing es un lenguaje de programación y entorno de desarrollo integrado de código abierto basado en Java, de fácil utilización, y que sirve como medio para la enseñanza y producción de proyectos multimedia e interactivos de diseño digital. Fue iniciado por

Ben Fry y Casey Reas a partir de reflexiones en el Aesthetics and Computation Group del MIT Media Lab dirigido por John Maeda. Se distribuye bajo la licencia GNU GPL.

```

/*
  Blink
  Turns on an LED on for one second, then off for one second, repeatedly.

  This example code is in the public domain.
*/

void setup() {
  // initialize the digital pin as an output:
  pinMode(13, OUTPUT);
}

void loop() {
  digitalWrite(13, HIGH); // set the LED on
  delay(1000);           // wait for a second
  digitalWrite(13, LOW); // set the LED off
  delay(1000);           // wait for a second
}

```

Ejemplo de programación con Processing

Sin embargo debido al enfoque de este proyecto, el lenguaje de programación para Arduino va a ser Scratch a través de S4A. Lo que facilita todavía más su programación y el desarrollo de aplicaciones.

5.3 SCRATCH 4 ARDUINO (S4A)

S4A es una modificación de Scratch que permite programar la plataforma de hardware libre Arduino de una forma sencilla. Proporciona bloques nuevos para tratar con sensores y actuadores conectados a una placa Arduino.

Gracias a esta herramienta se va a poder aprovechar en el segundo proyecto ABP, diseñado en este TFM, lo aprendido en el primer proyecto con Scratch para poder programar Arduino con S4A ya que esta basado en este último.

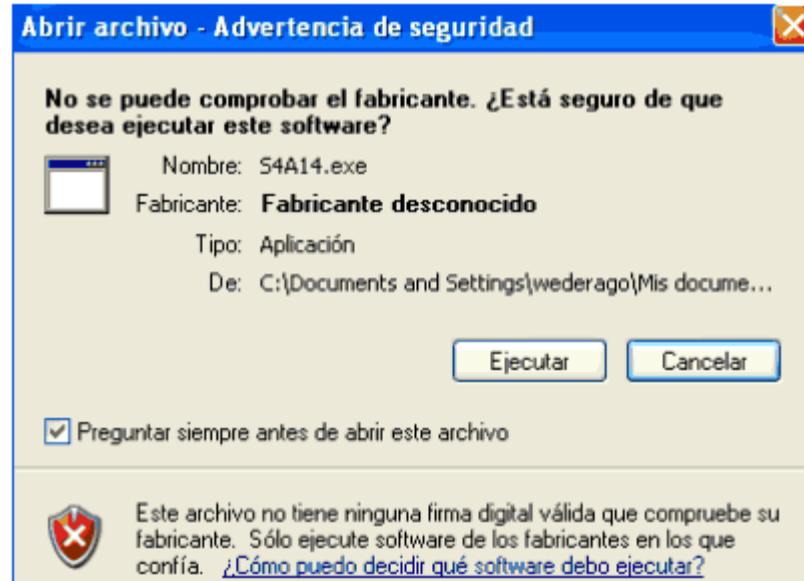
Instalación de S4A

Para hacer uso de este software necesitamos tener instalado el entorno de Arduino. Este paso ya ha sido explicado en el apartado anterior por lo que procederemos a la instalación de S4A.

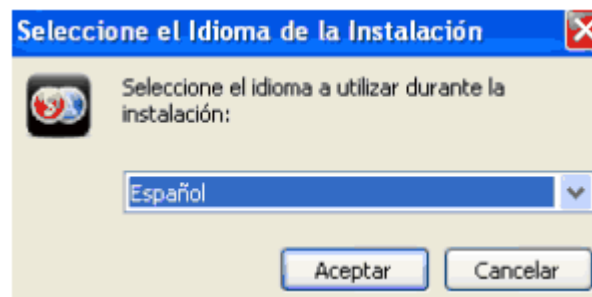
En la web del proyecto, en el apartado de descargas (downloads) encontraremos instaladores para Windows, Mac y Linux.

En nuestro caso procedemos a descargarnos el de Windows y lo ejecutamos.

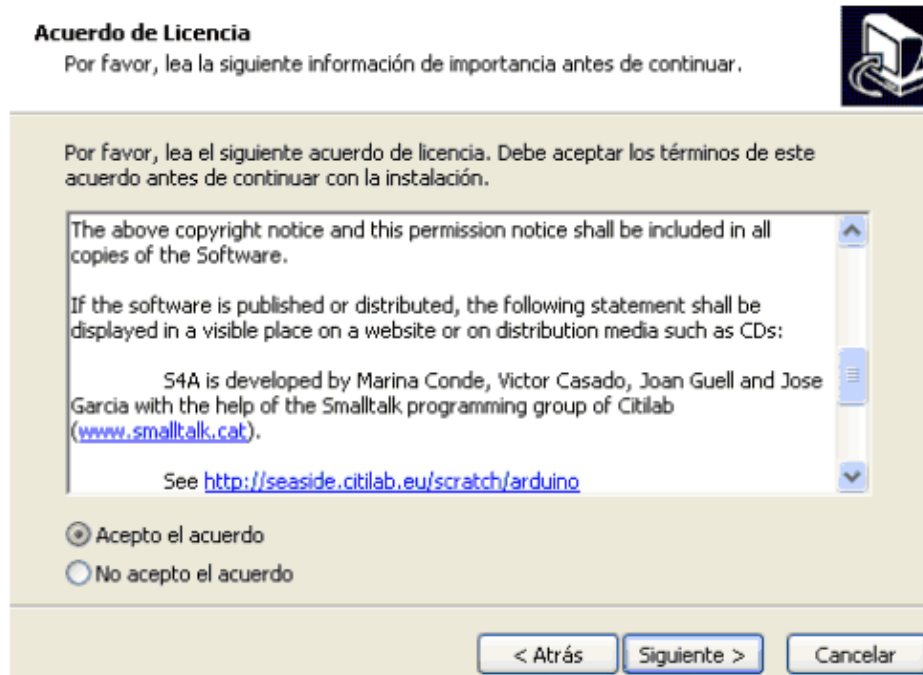
En caso de aparecer lo siguiente le daremos a ejecutar:



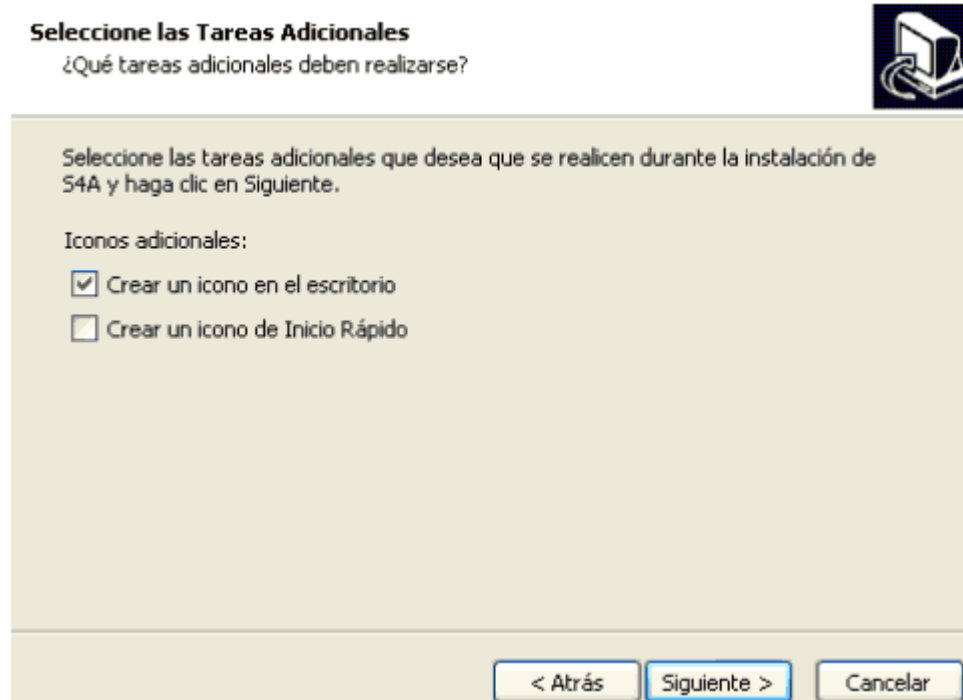
Seleccionamos idioma y procedemos a su instalación.



En las siguientes pantallas elegimos siguiente y aceptamos los términos de uso.



Seguimos presionando siguiente y elegimos si queremos crear algún tipo de acceso directo.



Ya tenemos instalado el software S4A. Ahora necesitamos que Arduino sea capaz de mandar datos a Scratch al mismo tiempo que los recibe.

Para ello de la misma web de Citilab nos descargamos el programa de Arduino que se encarga de ello asignando a los PIN la funcionalidad anteriormente descrita.

Descargamos el firmware del paso 2 que se indica en la web.

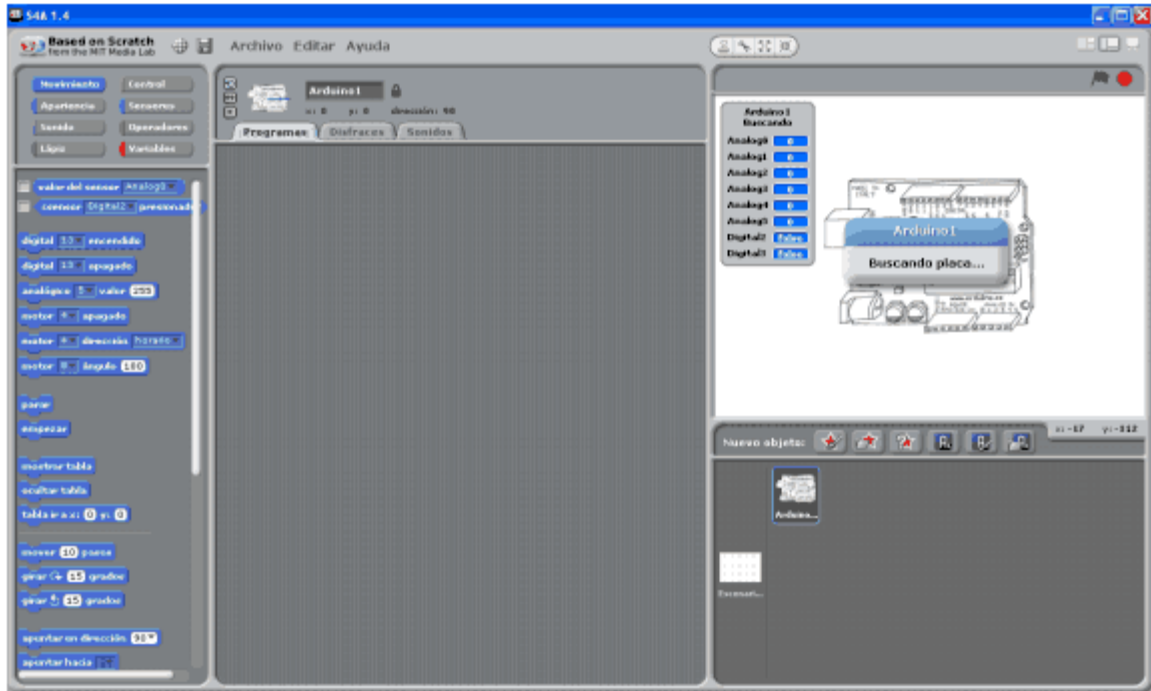
http://seaside.citilab.eu/scratch?_s=ZXxID7E_t-r92kJ&k=04llCkg5bVRVUPf6

Una vez descargado tan solo tenemos que cargar el archivo en la placa y ya podremos establecer la comunicación entre el hardware y el software.

Como ya tenemos subido el Sketch a la placa podemos cerrar el entorno de Arduino y comenzar a trabajar con S4A.

Al lanzar S4A existe la posibilidad de que nos solicite un archivo .image. Este fichero lo encontraremos en la misma ruta en la que instalamos S4A con el nombre S4A.image. Lo seleccionamos y ya estará listo S4A.

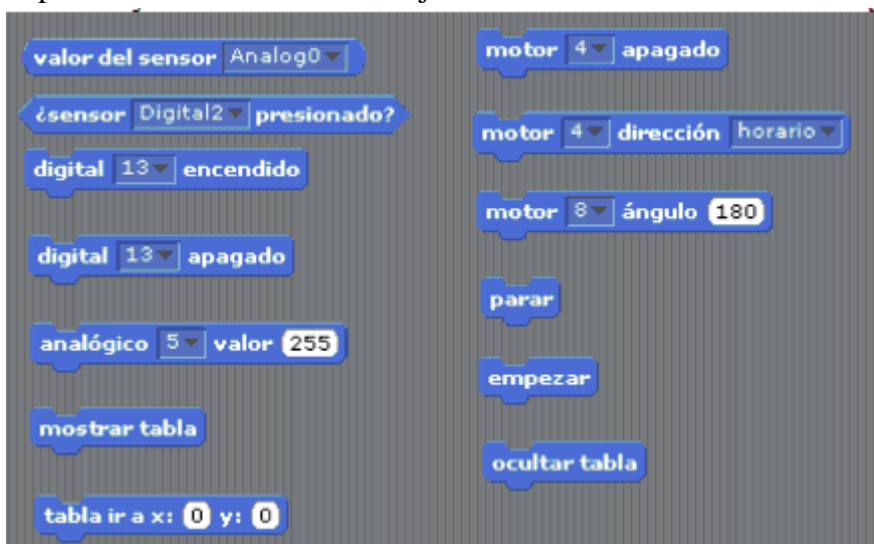
El interfaz de S4A es muy similar al de Scratch, pero añadiendo bloques que permiten enviar y recibir datos desde Arduino.



Durante los primeros segundos de S4A buscará una placa compatible para trabajar. Al encontrarla aparecerá:



Si miramos dentro del entorno del programa los bloques, podremos apreciar que casi todos coinciden con la programación de Scratch pero además tenemos unos cuantos nuevos que corresponden con la parte de interconexión con tarjeta Arduino.



Los bloques, “valor del sensor X” y “¿sensor X presionado?”, son similares a los

que encontramos en Sensores y nos permiten comprobar el estado de los PIN digitales y analógicos. Además de bloques exclusivos para el uso de PIN digitales y analógicos.

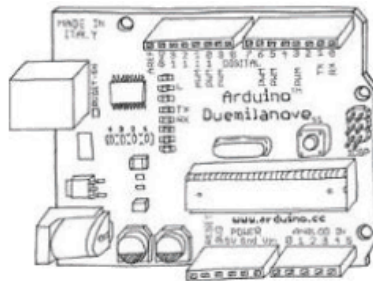
Por último encontraremos los encargados de los motores.

Los bloques de “ocultar tabla”, “mostrar tabla” y “tabla ir a: x, y” hacen referencia a la tabla que posee cada uno de los “Objetos Arduino” que tengamos en nuestro programa de Scratch.

Arduino 1	
puerto: COM4	
Analog1	195
Analog2	193
Analog3	190
Analog4	191
Analog5	194
Analog6	165
Digital1	false
Digital2	false

Un “Objeto Arduino” es un sprite que tiene acceso pleno a la comunicación entre Scratch y Arduino, es decir, posee todos los bloques que permiten la conexión.

Siempre que empezamos un proyecto nuevo tendremos un objeto Arduino por defecto con el siguiente disfraz.



En caso de necesitar más sprites que tengan acceso a estos datos, podemos hacer uso de los nuevos iconos que han aparecido a la derecha de los botones que usábamos para crear nuevos sprites en Scratch.



Los nuevos botones que tenemos nos permiten, de izquierda a derecha:

1. Crear un nuevo objeto Arduino: Crear un nuevo objeto con el mismo disfraz que nuestro primer objeto Arduino.
2. Dibujar objeto Arduino: Nos da acceso a las herramientas de dibujo de Scratch para que creamos nuestro objeto Arduino al igual que haríamos con cualquier otro.
3. Escoger desde fichero: Con esta opción tomamos uno de los sprite predefinidos o importados que tengamos y le da acceso a los datos y bloques exclusivos de Arduino.

Ya estamos familiarizados con las nuevas funciones de las que disponemos ahora, vamos a crear nuestra primera aplicación.

Primer contacto con S4A

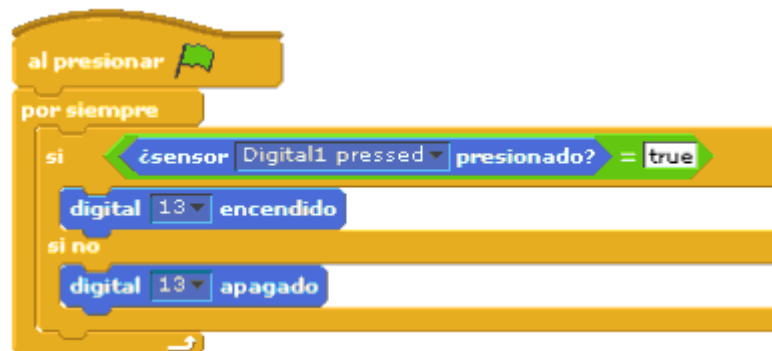
Como primer ejemplo con S4A se va a ver un pulsador que acciona un LED. Para ello antes se debe montar en la placa Arduino el circuito con pulsador y Led de salida.

Una vez tenemos el circuito montado abrimos S4A y comenzamos creando un bloque que compruebe en todo momento un valor, esto lo hacemos con un bloque “por siempre” y un “si”. Para este ejemplo usaremos el “si” que nos permite ejecutar dos bloques en función de si se da la condición o no. Y no nos olvidemos del bloque “al presionar empezar”

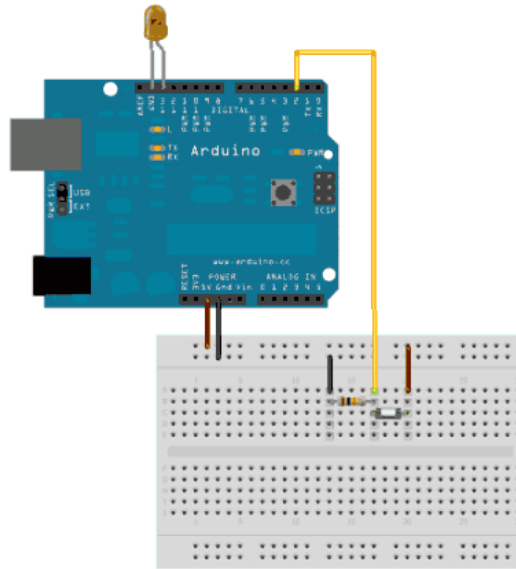


Queremos gobernar una salida digital PIN 13 mediante el accionamiento de una entrada digital PIN 2 a la que le hemos colocado un pulsador.

El programa es muy sencillo. Mediante el bloque de “sensor... presionado” averiguamos si el valor de la entrada es “true” o “false” y en función de este, mediante el bloque de función “si... si no” perteneciente a la librería “control”, conseguimos la función de gobierno deseada: Activar o desactivar la salida PIN 13



El esquema de montaje es el de la figura:



Conclusión

Con el ejemplo anterior hemos visto cómo, de manera sencilla, podemos hacer en S4A lo que podríamos hacer usando Processing sin muchos conocimientos del entorno de Arduino. Haciendo uso de la programación intuitiva y gráfica de Scratch tenemos acceso a todo el mundo de posibilidades que nos brinda Arduino, eso sí, con la limitación en el número de entradas y salidas de las que dispone la placa y están preconfiguradas en S4A.

6. APRENDIZAJE BASADO EN PROYECTOS CON SCRATCH Y S4A+ARDUINO.

Como se ha comentado al inicio de este TFM el objetivo es crear material para la asignatura de Tecnología de 4 ESO, en la cual se puedan desarrollar proyectos a través de la metodología ABP e introducir a los alumnos en una edad relativamente temprana a la programación o computación a través de la herramienta Scratch. Se han planteado a continuación dos proyectos con metodología ABP, los cuales están relacionados entre sí a través del tema sobre **el calentamiento global o cambio climático**. Es un tema que necesita de la máxima sensibilización por parte de todos para poder construir un mundo más sostenible, por eso he utilizado este tema como hilo conductor de los proyectos aquí presentados.

Con el primer proyecto se trata de profundizar en diferentes consecuencias del cambio climático para diseñar la temática del juego y con el segundo proyecto se trata de promover acciones que supongan un beneficio medioambiental.

Por tanto los dos ABP se titulan:

- ABP 1: Videojuego sobre el cambio climático con Scratch.
- ABP 2: Eficiencia energética en el centro educativo utilizando S4A+Arduino.

ABP 1 APRENDIZAJE BASADO EN PROYECTOS

VIDEOJUEGOS CON SCRATCH CAMBIO CLIMÁTICO



6.1 Videojuego cambio climático con Scratch

1. INTRODUCCIÓN:

Este primer ABP está enfocado a desarrollar las competencias en computación mediante la introducción a la programación con Scratch. Se va a llevar a cabo este trabajo en sincronización con la asignatura de ciencias naturales ya que se va a utilizar el tema del cambio climático para desarrollar un videojuego con Scratch. Esta temática es desarrollada en su currículo y servirá para poder profundizar posteriormente en el desarrollo del proyecto. De manera libre el grupo elegirá un tema o problemática relacionada con el cambio climático para desarrollar el escenario donde va a ocurrir el videojuego o historia. A modo de ejemplo señalamos posibles temas para el proyecto.

- Antártida
- Aumento de las temperaturas, desertificación.
- Tala masiva incontrolada.
- Pingüinos, osos polares escapando del deshielo
- Inundaciones, huracanes.
- Reciclaje
- Ahorro energía

1.1 Contexto

El proyecto esta pensado para un grupo de alumnos de 4º ESO y se pretende cubrir el bloque 5 y 7 del curriculum de Tecnológica y el bloque 5 del currículum de Ciencias Naturales, según el Decreto Foral 25/2007 para estas dos asignaturas.

Tecnología:

Bloque 5. Control y robótica.

- **Experimentación con sistemas automáticos, sensores, actuadores y aplicación de la realimentación en dispositivos de control.**
- Diseño y construcción de robots.
- **Uso del ordenador como elemento de programación y control. Trabajo con simuladores informáticos para verificar y comprobar el funcionamiento de los sistemas diseñados**

Bloque 7. Tecnología y sociedad.

- Aprovechamiento de materias primas y recursos naturales.
- Adquisición de hábitos que potencien el desarrollo sostenible.

Ciencias Naturales:

Bloque 5. La contribución de la ciencia a un futuro sostenible.

- Un desarrollo tecnocientífico para la sostenibilidad:

- Los problemas y desafíos globales a los que se enfrenta hoy la humanidad: contaminación sin fronteras, cambio climático, agotamiento de recursos, pérdida de biodiversidad, etc.
- Contribución del desarrollo tecnocientífico a la resolución de los problemas. Importancia de la aplicación del principio de precaución y de la participación ciudadana en la toma de decisiones.
- Valoración de la educación científica de la ciudadanía como requisito de sociedades democráticas sostenibles.
- La cultura científica como fuente de satisfacción personal

1.2 Metodología

La metodología es la de ABP y se va a utilizar en la asignatura de Tecnología.

Se basa en los siguientes puntos:

- El aprendizaje se construye en base a los conocimientos previos que tienen los alumnos (constructivismo, Piaget)
- Los alumnos aprenden trabajando e investigando en grupo (indagación).
- El profesor realiza el papel de tutor acompañando a los alumnos, su papel está muy lejos del profesor magistral (constructivismo, vigotsky).
- Se aprende en base a problemas cercanos al alumno y que cubre el currículo de su formación (trabajo de contenidos y de competencias).
- El contexto del aprendizaje es esencial en la formación del alumno (inmersión en la estrategia PBL e identificación con los problemas a resolver).

Dentro de clase se presenta el problema que los estudiantes deben resolver, se discuten las reglas, las probables soluciones, el tiempo máximo del proyecto, limitación de materiales, cómo se debe presentar la solución del problema.

Durante la ejecución del proyecto el profesor se convierte en un facilitador, es decir, a solicitud del estudiante actúa como consejero, para ayudar que el estudiante descubra e investigue por sí mismo. Permitiendo que el estudiante actúe con la máxima libertad posible.

Cada grupo presenta su proyecto de acuerdo a las bases de cada proyecto, se revisa la documentación y se realizan la prueba.

1.3 Objetivos

Los objetivos con esta metodología son:

- Inculcar la organización y la importancia del trabajo en equipo y potenciar actitudes flexibles y responsables, en la toma de decisiones, ejecución de tareas y búsqueda de soluciones basándose en la cooperación, imaginación y creatividad.

- Abordar con autonomía y creatividad problemas tecnológicos sencillos trabajando de forma ordenada y metódica (diseñar y construir objetos o sistemas que resuelvan el problema estudiado, y evaluar su idoneidad), basándose en el método de indagación.
- Desarrollar interés y curiosidad hacia la actividad tecnológica, generando iniciativas de investigación, así como de búsqueda y elaboración de nuevas realizaciones tecnológicas.
- Estimular la capacidad personal de aprendizaje, basándose en los aspectos más importantes del aprender a aprender y promover la responsabilidad por el aprendizaje propio.
- Aumentar el uso de las TIC para solucionar problemas tecnológicos.
- Amenizar la asignatura impartiendo clases más dinámicas y participativas, llevando a la práctica los contenidos teóricos y relacionándolos con la realidad.

A través de este proyecto se pretende además que los alumnos puedan adquirir los conocimientos relacionados con los diferentes bloques de contenidos a trabajar:

Tecnología:

- Contribuir al desarrollo de habilidades del siglo XXI, capacidades intelectuales de orden superior y pensamiento computacional, por parte de estudiantes.
- La programación de computadores posibilita no solo activar una amplia variedad de estilos de aprendizaje sino desarrollar pensamiento algorítmico.
- La programación se convierte en una estrategia válida para implementar el enfoque educativo conocido con la sigla STEM (Ciencia, Tecnología, Ingeniería y Matemáticas). Con este enfoque, las naciones industrializadas buscan preparar en estas áreas una cantidad suficiente de docentes y estudiantes para responder adecuadamente a los tiempos actuales de globalización e innovación, donde gran parte de los nuevos puestos de trabajo los genera el sector de las Tecnologías de la Información y la Comunicación (TIC).
- Preparar a los estudiantes desde la primaria, para que puedan insertarse activamente al mundo altamente programado que los espera.
- Aplicar los elementos básicos del lenguaje SCRATCH al diseño de programas que faciliten los aprendizajes de la electrónica básica.
- Usar los comandos de apariencia y sonido del entorno scratch con el fin de elaborar programas fundados en el aprendizaje de la robótica educativa que resulten entretenidos y creativos.
- Utilizar los elementos básicos del lenguaje SCRATCH al diseño de programas que faciliten los aprendizajes de la robótica.

1.4 Contribución a la adquisición de competencias básicas

El trabajar con metodología ABP esta íntimamente relacionado con el trabajo por competencias, además de con la herramienta Scratch que impulsa y refuerza la utilización de muchas de ellas. Se van a enumerar las ocho competencias básicas, las cuales se van a poder

trabajarse en este ABP.

1. Conocimiento e interacción con el mundo físico.

–Realiza pequeñas investigaciones sobre diferentes fenómenos reflejando interpretaciones cualitativas en un informe sencillo. Diseña, planifica y monta aparatos formados por elementos simples

2. La competencia matemática está íntimamente asociada

–Maneja con fluidez los números resolviendo y comprobando la validez del resultado en problemas de su vida real.

–Identifica y describe regularidades y tiene destreza para resolver problemas algebraicos sencillos.

– Reconoce figuras geométricas en su entorno directo, las clasifica según propiedades, estima y calcula longitudes, superficies y volúmenes para resolver retos geométricos en la elaboración de maquetas.

3. Tratamiento de la información y competencia digital

–Busca y selecciona información en distintas fuentes y soportes, la organiza y contrasta de forma guiada para realizar sencillas investigaciones que redacta aplicando modelos de elaboración de informes y trabajos de síntesis, valorando la importancia de su planificación.

–Utiliza Internet, redes interactivas y herramientas de comunicación con fines educativos o de relaciones sociales, mediante el intercambio de información, realización de trabajos cooperativos, uso de foros de dudas, de temas, discusión y envío de actividades y tareas.

4. Aprender a aprender

– Atribuye sus resultados a causas controlables y se muestra dispuesto a modificar sus estrategias para conseguir sus metas, identificando lo que sabe y lo que le queda por aprender.

–Se hace preguntas sobre los objetivos, pasos y procedimientos para planificar y realizar las diferentes tareas, revisándolas con pautas de autocorrección y comienza a pensar en alternativas de mejora.

5. Competencia social y ciudadana

–Establece relaciones de respeto en el trabajo cooperativo aceptando las responsabilidades propias del rol desempeñado

6. Comunicación lingüística

–Lee textos periodísticos, literarios y técnicos, redactando en distintos soportes mensajes con claridad para compartir sus ideas y conocimientos

–Selecciona adecuadamente sus palabras al expresar pensamientos y emociones y participa abiertamente en debates y exposiciones escolares explicando ordenadamente sus conocimientos sobre sucesos observados y temas conocidos

7. Autonomía e iniciativa personal

–Muestra un nivel aceptable de seguridad y confianza que se refuerza ante sus iguales, expresando sus ideas y aceptando las contribuciones de los otros en el desarrollo de su función en equipos cooperativos

8. Competencia cultural y artística

–Comunica mensajes con imaginación y creatividad, de forma individual o grupal, mediante producciones sencillas, eligiendo los materiales e instrumentos de uso escolar adecuados, incluidos los relacionados con las tecnologías de la información y comunicación.

En la secuenciación de las sesiones para realizar el proyecto se refleja que competencias se trabajan en cada sesión.

2. DESCRIPCIÓN DEL PROYECTO

2.1 Jugar creando: Cambio climático-Calentamiento Global de la Tierra

Reto, pregunta guía o hilo conductor: «El calentamiento del sistema climático es inequívoco, como ya es evidente de las observaciones e incrementos en el promedio de la temperatura del aire y del océano, la fundición de nieve y hielo, y el incremento del nivel medio del mar a nivel mundial. Y con un 90% de certeza es debido a la mano del ser humano»

Son numerosas las consecuencias que el cambio climático está teniendo y va a tener sobre nuestro planeta. Dichos efectos no inciden por igual en todas las regiones del mundo, siendo las más desfavorecidas y densamente pobladas las más vulnerables.

Partiendo de esta base, el videojuego o historia desarrollada con la plataforma Scratch debe contener esta temática sobre el cambio climático o calentamiento global del planeta.

Con las actividades que se proponen en esta actividad se pretende concienciar en que todos somos parte del problema, y a la vez, la solución al mismo.

El proyecto consta de tres partes principalmente:

- En la asignatura de ciencias de la naturaleza se analizará el tema del cambio climático y sus repercusiones. De manera coordinada se impartirá ese temario antes de realizar el ABP en tecnología. También se realizará un seminario con un experto en la materia en una sesión de tecnología para coordinar esfuerzos y dudas del alumnado para realizar el proyecto.
- Una fase de toma de contacto y profundización con Scratch de manera individual para fomentar destrezas antes de desarrollar el proyecto final.
- Desarrollo de un videojuego o historia interactiva contextualizada en dicha problemática mediante la metodología ABP de manera grupal.

Además el alumnado debe desarrollar una presentación en Power Point o Prezi describiendo la problemática climática que ha elegido, el porqué y cuáles son sus efectos. También se mostrará el desarrollo de diseño desde la idea hasta los personajes, el reto del juego, así como la presentación del juego a través de su publicación en la web social Scratched para poder ser vista y jugada por todos.

Para la realización del proyecto se hará hincapié en la secuencia: Plan – Comprobación – Modificación a la hora de obtener un buen resultado.

Para terminar el concurso el alumnado se autoevaluará su creación mediante la rúbrica creada en Scratch. Una vez hayan finalizado sus videojuegos, se los presentarán a los alumnos de 3º de ESO en la asignatura de Tecnológica.

Ejemplos de videojuego o historias:

- <http://www.oletusjuegos.es/juegos-de-calentamiento-global-online.htm>

Descripción: El calentamiento global está derritiendo todo el hielo de La Antártida, los pingüinos se han dado cuenta y quieren escapar antes de que sea tarde.

Ayuda a Peter el pingüino a escapar del continente helado.

- [Videojuego cambio climático](#)
- <http://scratch.mit.edu/projects/11874988/>
- <http://scratch.mit.edu/projects/19692559/>

2.2 Objetivos

El estudiante debe presentar y producir un videojuego, historia usando la plataforma Scratch 2.0.

Se deben entregar dos productos, la planeación y el producto final:

1. La planeación es una presentación hecha en PowerPoint para exponer el día de la entrega final y debe contener, como mínimo, los siguientes contenidos:

- El juego debe tener un nombre único
- El juego debe tener una historia que describe la interacción entre los varios elementos de que la componen:
 - Personajes (protagonista, antagonicos y extras)
 - Contexto (época, lugar, etc.)
 - Las reglas y mecánica del juego (objetivo y que puede hacer el jugador para lograrlo). Debe quedar muy claro cómo ganar y como perder.

2. El producto final es un juego hecho en Scratch que debe tener como mínimo los siguientes elementos:

- Diseño y creación de por lo menos un personaje y un escenario (se evalúa la calidad)
 - Cada personaje creado debe tener múltiples disfraces para mostrar todos sus movimientos
- Dos niveles de dificultad, siendo el segundo más complicado de lograr que el primero.
- Manejo de efectos de sonidos y música de fondo
 - Grabar por lo menos uno de los efectos
- Manejo y visualización de puntajes durante todo el juego

2.2 Duración

El proyecto durará 7 semanas con 21 sesiones, ya que cada semana consta de tres sesiones de tecnología de 4º ESO.

De esas 21 sesiones 13 se trabajarán de manera individual, donde el profesor realizará labores de acompañamiento en el proceso de aprendizaje con Scratch y las últimas 8 sesiones (casi tres semanas) se destinarán a la elaboración del juego en grupos de 4 personas.

Las sesiones presentadas en este TFM están organizadas dentro de 5 partes para que los estudiantes puedan explorar diferentes géneros de expresión creativa y originales, mientras de manera fluida y a la vez lúdica se desarrollan conceptos computacionales.

2.3 Los grupos

La clase se dividirá en 5 grupos de 4 personas. El profesor formará los grupos, teniendo en cuenta la relación entre dichos alumnos.

Aunque a cada alumno se le asigna una actividad dentro de cada grupo, seguirán vigentes las responsabilidades que en el aula se hayan establecido para la materia de tecnología:

Dentro de cada grupo habrá distintos responsables de la organización del trabajo :

- Responsable de material y limpieza: Este alumno se encargará de llevar el material desde el almacén a la zona de trabajo. Además Al finalizar la clase, será quien recoja la mesa de trabajo y quien lleve los restos de material a la papelería.
- Secretario: Es la persona responsable de guardar toda la información del proyecto en la carpeta correspondiente (la memoria, croquis,...) y avisar a todos los miembros del grupo de los días en los que quedarán para realizar el proyecto.
- Cronometrador: es la persona responsable en controlar el tiempo necesario para la ejecución de cada actividad. De esta manera se gana en eficacia y permite avanzar en las actividades.
- Responsable TIC's y para este proyecto de la web social: <http://scratch.mit.edu/>. Se encargará de subir las evoluciones del proyecto a la web social para poder compartir con el resto de la clase las evoluciones del proyecto y demás cuestiones sobre la web.

Cada grupo entregará al profesor un documento con los nombres de los participantes, los diferentes roles que van a desempeñar en dicho grupo y las tareas que cada miembro va a realizar en la creación del juego (creación de Sprites, escenarios, lista variables, definición de la historia, etc.).

3. DESARROLLO DEL PROYECTO

La temporización elegida para este PBL será de 8 semanas y 24 sesiones en total.

De cara a relacionar las sesiones con las competencias básicas que se trabajarán se va a enumerar dichas competencias básicas y en las sesiones colocar el correspondiente número con la que se trabaja.

	COMPETENCIAS BÁSICAS
1	Competencia en comunicación lingüística.
2	Competencia matemática.
3	Competencia en el conocimiento y la interacción con el mundo físico.
4	Tratamiento de la información y competencia digital.
5	Competencia social y ciudadana
6	Competencia cultural y artística.
7	Competencia para aprender a aprender.
8	Autonomía e iniciativa personal.

Durante el transcurso del proyecto se marcan varios hitos donde los grupos tendrán que entregar parte del trabajo y hacer una pequeña presentación de lo hecho hasta el momento. Todas las tareas que se vaya a realizar se distribuyen de la siguiente manera:

	Modo de trabajo	Descripción	Nº de sesiones
Introducción	Individual	-Explicación del proyecto. -Los estudiantes se introducen en la programación con Scratch mediante diferentes ejemplos y manejo de experiencias.	2
Seminario	Individual	Junto con profesor de Ciencias de la Naturaleza los estudiantes exploran diferentes ideas para el videojuego.	2
Programación básica	Individual	Con el tópic del cambio climático como hilo conductor desarrollaremos pequeños programas que incluyen muchos de los bloques de control de Scratch.	1
Historias	Individual	Los estudiantes exploran diferentes historias relacionadas con el cambio climático mediante la creación de	3

		personajes, escenas y narrativas.	
Juegos	Individual	Los estudiantes exploran diferentes juegos con la creación de proyectos que definen objetivos y reglas.	4
Proyecto final	Grupos de cuatro alumnos/as	Los estudiantes en grupo de cuatro realizan proyectos independientes mediante la definición de un proyecto en el que trabajar, colaborar con los demás en mejorar el proyecto y presentar dicho proyecto así como su desarrollo de trabajo y evaluación de los demás videojuegos.	10

Sesión 1 y 2

- Descripción:

Explicación del proyecto y forma de trabajar, tareas a desarrollar.

¿Qué es programar?

Introducción a los alumnos al entorno Scratch, , ejemplos con Scratch, creación de cuenta Scratched (web social) y exploración inicial de pequeños proyectos con Scratch.

- Objetivos:

-Entender y ver las posibilidades de la herramienta Scratch.

-Conocer Scratch y sus creaciones de manera lúdica.

-Construir y manejar los primeros bloques con Scratch

-Ser capaces de comenzar a moldear los bloques de Scratch para sus propios intereses de creación.

-Compartir y discutir las diferentes creaciones entre al alumnado y con el profesor.

- Actividades:

Definición de programación y algoritmos en la vida real

Introducir a los estudiantes dentro del entorno Scratch

Mostrar diferentes ejemplos que contiene el propio Scratch e intentar comprenderlos.

Revisar diferentes procesos sencillos de creación para la programación con Scratch.

Descarga y instalación de Scratch en sus ordenadores.

Creación cuenta en su Web social.

- Competencias: 1, 3, 4, 5, 6, 8

Sesión 3 Seminario con experto en calentamiento global.

- Descripción: Se realizará un seminario informativo por parte de un experto en calentamiento global y el profesor de ciencias naturales. De esta manera el alumnado explorará diferentes ideas o tópicos que pueden ser objeto de narración para el proyecto a realizar al final del proyecto.

- Objetivos:

-Recordar y desarrollar ideas que sirvan de hilo para realizar el videojuego o historia con Scratch.

-Concienciación sobre los problemas del cambio climático.

- Actividades:

-Debate inicial para reconocer las ideas previas sobre el tema a tratar.

-Exposición de los expertos sobre la temática.

-Tarea sobre visionado de videos con temática del calentamiento global como la desertización, falta de agua, deshielo polar, aumento inundaciones y huracanes, tala masiva de árboles. Para ello veremos el documental [*Home, historia de un viaje*](#)

- Competencias: 1, 3, 5, 7, 8

Debatiremos al final sobre lo más impactante o que más nos ha sobrecogido para que puedan posteriormente enfocar sus esfuerzos en la realización del videojuego en torno a esa idea.

Sesión 4 y 5

- Descripción: Comenzar a generar las primeras animaciones e historias

En esta sesión el alumnado explorará conceptos de instrucción y secuenciación a través del eje central del proyecto: El cambio climático.

Para ello el alumnado tendrá que explicar, valiéndose de Scratch, en qué consiste el cambio climático.

La puesta en común posterior permitirá reflexionar sobre las diferentes dimensiones del problema (causas, acciones humanas, consecuencias, teorías científicas, alternativas y soluciones, etc.). Además, se podrán clarificar dudas, concepciones erróneas e incorporar nuevos datos sobre el cambio climático.

- Objetivos:

-Aprender a expresar actividades más complejas a través de instrucciones.

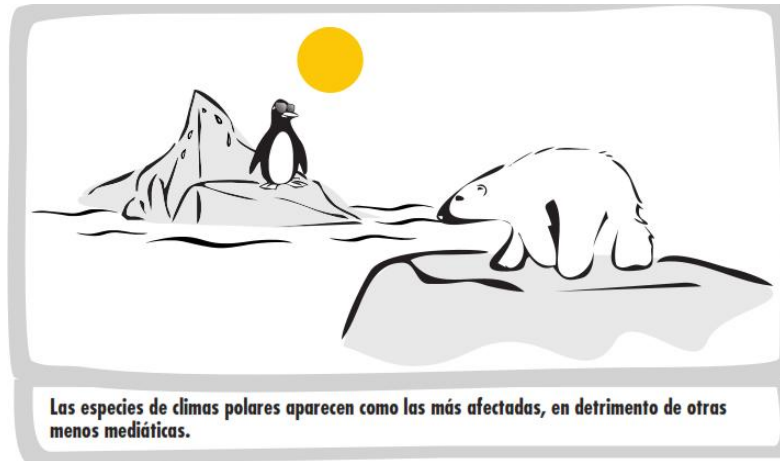
-Ser capaces de combinar animación y música.

- Actividades:

-Debate inicial para reconocer las ideas previas sobre el tema a tratar.

Trabajo para entregar-->Se les leerá el siguiente texto: Gurb es un marciano llegado de otro planeta. La ventana temporal que le ha permitido viajar a la Tierra sólo estará abierta durante 20 minutos. Tú eres la primera persona con la que se encuentra. Viene a buscar información sobre el cambio climático y te pide que le hagas un dibujo, un esquema, representación gráfica que tú quieras para poder llevarla a su planeta y explicar allí en qué consiste ese fenómeno. Deberá contener algun elemento dinámico.

-Utilización de diferentes bloques de control, sonido e interacción entre Sprites.



- Competencias: 1, 2, 3, 4, 5, 6, 7, 8

Sesión 6

- Continuaremos con los diferentes bloques de programación más complejos.
- Objetivos

-Desarrollar una mayor fluidez con los conceptos computacionales (es decir, secuencias, loops, eventos) y prácticas (es decir, el desarrollo iterativo e incremental, pruebas y depuración, la reutilización y la remezcla, resúmenes y modularización) trabajando en un proyecto auto-dirigido

- Actividades

-Debate inicial para reconocer las ideas previas sobre el tema a tratar.

-Utilización de bloques de control sensores, variables y operadores de Scratch. Para ello aumentaremos el proyecto ya iniciado en la sesión anterior introduciendo instrucciones donde se interactúa con el jugador a través de preguntas y respuestas.

Se deberán introducir diferentes cuestiones como:

1. ¿La Antártida es más grande que Europa?	Sí	No
2. ¿Si se derrite todo el hielo de la Antártida se espera que el nivel del mar llegue a subir más de 50 metros?	Sí	No
3. ¿Hay tierra, ríos, lagos y montañas debajo del hielo en la Antártida?	Sí	No
4. ¿En la Antártida nieva y llueve más que en tu ciudad?	Sí	No
5. ¿La Península Antártica es uno de los lugares de la Tierra donde más ha aumentado la temperatura en los últimos 50 años?	Sí	No
6. ¿Con el calentamiento global habrá más osos polares que pingüinos en la Antártida?	Sí	No
7. ¿Ha habido alguna vez árboles en la Antártida?	Sí	No
8. ¿Hay un agujero en la capa de ozono sobre la Antártida?	Sí	No
9. ¿Los dueños de la Antártida son los esquimales?	Sí	No
10. ¿Existe algún acuerdo internacional para proteger y conservar la Antártida?	Sí	No

-Entrega al profesor de creación realizada para evaluación.

- Competencias: 1, 2, 3, 4, 5, 6, 7, 8

Estas sesiones de diseño proporcionan una oportunidad para llegar con los estudiantes que puedan necesitar alguna atención o apoyo adicional.

Sesión 7 y 8 Historias

- Descripción:

Comenzaremos con la creación de historias.

En esta sesión, los estudiantes exploran los conceptos de paralelismo y eventos a través del desempeño y la historia.

- Objetivos

-Ser capaz de explicar lo que es el paralelismo y cómo funciona en scratch

-Ser capaz de explicar qué son eventos y cómo funcionan en el scratch

-Comprender los beneficios del remezclar mientras se diseña.

-Ser capaz de crear un proyecto de Scratch que cuenta una historia basándose en el trabajo de otros.

- Actividades

-Debate inicial para reconocer las ideas previas sobre el tema a tratar(combustibles fósiles)

-Escribe historias de seis palabras sobre el tema de los combustibles fósiles.

-Diseñar y comenzar a construir una historia de un mundo sin combustibles fósiles. Para ello se deben utilizar con Scratch actividades paralelas y orientadas a eventos.

-Crear colaborativamente proyectos de Scratch historia a través de la remezcla

-Compartir y discutir creaciones

- Competencias: 1, 2, 3, 4, 5, 6, 7, 8

Varias ideas de peso se exploran en esta actividad. En primer lugar, la noción de reposición es algo Scratchers luchan nada más empezar. Se programa todo de cero y si quieres que las cosas comienzan en un lugar determinado, con una orientación particular, etc, cada uno es el único responsable de los pasos de para su configuración. En segundo lugar, hay múltiples niveles de paralelismo con Scratch. Un solo sprites puede hacer varias cosas a la vez, y múltiples sprites también puede realizar acciones de forma simultánea. Ser capaz de leer el código de los demás es una habilidad valiosa y es fundamental para poder participar en el prácticas de reutilización y remezclar.

Sesión #9 Historias

- Descripción:Continuar con la historia planteada o diseñada en sesión anterior pero incorporando nuevos bloques, escenarios, etc.

- Objetivos:

-Desarrollar una mayor fluidez con los conceptos computacionales (es decir, el paralelismo, eventos) y prácticas (es decir, el desarrollo iterativo e incremental, pruebas y depuración, la reutilización y la remezcla, resúmenes y modularización) trabajando en un proyecto auto-dirigido.

- Actividades

-Trabajar en proyectos de Scratch

-Entrega al profesor de historia realizada para evaluacion.

- Competencias: 1, 2, 3, 4, 5, 6, 7, 8

Notas

Esta sesión de diseño abierto proporciona una oportunidad para llegar a los estudiantes que puedan necesitar alguna atención o apoyo adicional.

Sesión 10 y 11 Games

- Descripción: Comenzaremos con la creación de juegos con Scratch.
- En esta sesión, el alumnado explora una serie de conceptos (incluidos los condicionales y operadores) a través de las prácticas de pruebas y depuración. diseño de un laberinto.
- Objetivos

- Ser capaz de explicar las prácticas de prueba y depuración
- Desarrollar una lista de estrategias para los proyectos de prueba y depuración de Scratch
- Ser capaces de identificar algunos elementos comunes de diseño de juegos
- Ser capaz de utilizar Scratch para crear un juego de laberinto

- Actividades:

- Depurar varios proyectos de Scratch
- Diseño de un escenario de depuración
- Compartir y discutir estrategias de depuración
- Identificar los elementos de diseño de juegos comunes
- Crear un laberinto

- Competencias: 1, 2, 3, 4, 5, 6, 7, 8

Notas

Las pruebas y la depuración es probablemente la actividad más común de los programadores. Las cosas rara vez funciona según lo previsto, por lo que el desarrollo de un conjunto de estrategias de prueba y depuración será beneficioso para cualquier creador computacional.

Sesión 12 Games

- Descripción: En esta sesión, los estudiantes exploran los conceptos de condicionales y datos a través de la mecánica de juegos comunes.
- Objetivos

- Ser capaz de describir lo que una variable es y por qué las variables son útiles
- Actividades
- Desarrollar ampliaciones de los proyectos de laberinto
- Ayuda a otros a aprender acerca de las variables que utilizan una de las extensiones como ejemplo.

[Ejemplo laberinto](#)

- Competencias: 1, 2, 3, 4, 5, 6, 7, 8

Sesión 13 y 14 Games

- Descripción: Continuar con elaboración de un de un nuevo videojuego mas logrado.
- Objetivos

- Desarrollar una mayor fluidez con los conceptos computacionales (es decir, operadores condicionales, datos) y prácticas (es decir, el desarrollo iterativo e incremental, pruebas y

depuración, la reutilización y la remezcla, resúmenes y modularización) trabajando en un proyecto autodirigido

- Actividades

-Diseño de escenarios, Sprites para la creación del primer videojuego.

-Entrega al profesor de juego realizado para evaluación.

- Competencias: 1, 2, 3, 4, 5, 6, 7, 8

Nota: Esta sesión de diseño abierto proporciona una oportunidad para llegar con los estudiantes que puedan necesitar alguna atención o apoyo adicional, en particular alrededor de los conceptos de los condicionales (por ejemplo, if), operadores (por ejemplo, aritméticas, lógicas), y los datos (por ejemplo, las variables, las listas) .

Sesión 15 Proyecto Final

- Descripción: En esta sesión, los grupos se juntaran para trabajar en elección de temática calentamiento global para delinear sus proyectos finales.

- Objetivos

-Identificar un proyecto de manera apropiada.

-Desarrollar un esquema de las actividades o tareas necesarias para completar el proyecto

-Generar una lista preliminar de los recursos necesarios para completar el proyecto

- Actividades

-Creación de grupos de cuatro personas

-Proyectos finales Lluvia de ideas.

-Selección de funciones a realizar por cada miembro del grupo; creador Sprites, escenarios, diseñador historia o videojuego, manejo variables y puntacion.

-Planificación del proyecto.

-Trabajar en el proyecto final (si el tiempo lo permite)

-Recoger la planificación del proyecto.

Tomando algo de tiempo en el inicio del proyecto final para explorar ideas, identificar las tareas involucradas en la realización del proyecto, y la lista de lo que es (y no es) ya conocido puede ser muy beneficioso para una buena finalización del proyecto. Aunque la planificación es útil, no debería ser la única manera de hacer las cosas. El alumnado es diverso y quieren y necesitan planificar y jugar en distinta medida. Además las diferentes fases del proyecto pueden requerir diferentes enfoques. Diseño y desarrollo de múltiples estilos deben ser animados y acomodados.

- Competencias: 1, 2, 3, 4, 5, 6, 7, 8

Sesión 16 y 17 Proyecto Final

- Descripción: Puzzle y trabajo en parejas del proyecto final

- Objetivos

-Identificar las áreas en las que necesitan apoyo

-Proporcionar orientación y apoyo

- Actividades

-Reunión de los grupos con intereses especiales.

-Trabajar en proyecto final

Nota: Tener muchos estudiantes cada uno explorando diferentes caminos supone un reto interesante para el profesor - ¿cómo se apoya a un gran número de personas? Los estudiantes pueden ser de enorme valor en la prestación de apoyo y orientación a la otra a lo largo de todas las sesiones de Scratch, y particularmente durante las sesiones finales del proyecto. Dar al alumnado la oportunidad de enseñar otro hace las cosas más fáciles para el profesor, pero también puede profundizar significativamente el aprendizaje y la comprensión de los creadores.

- Competencias: 1, 2, 3, 4, 5, 6, 7, 8

Sesión #17 Proyecto Final

- Descripción: En esta sesión, los grupos trabajaran con otros grupos para recibir opiniones, críticas, es decir para dar y recibir retroalimentación sobre sus proyectos.

- Objetivos

-Prueba de proyectos en curso

-Formular y compartir retroalimentación para los demás grupos.

- Actividades

-Revisión elementos planificación

-Reunión de los grupos entre ellos.

-Los trabajos en proyecto final.

- Competencias: 1, 2, 3, 4, 5, 6, 7, 8

Notas: Diferentes personas ofrecerán diferentes perspectivas sobre el proyecto en curso. Crear oportunidades para los creadores para obtener retroalimentación de una variedad de fuentes, incluidos ellos mismos.

Sesión 18 y 19 Proyecto Final

- Descripción: Trabajar en proyectos finales y presentación del proyecto al resto de sus compañeros.

- Objetivos

Terminar de desarrollar un proyecto de Scratch de su elección.

Conocer todos los proyectos realizados por los diferentes grupos de clase.

- Actividades

-Los trabajos en proyecto final.

-Recogida de los trabajos por parte del profesor.

-Presentación de todos los grupos al resto de la clase sobre el trabajo elaborado.

- Competencias: 1, 2, 3, 4, 5, 6, 7, 8

Sesión 20 Evaluación grupal

- Descripción: Jugar con las creaciones de los demás grupos y autoevaluarse.

- Objetivos

-Disfrutar de los juegos creados y realización de la clasificación final mediante rúbrica creada con Scratch.

- Actividades

-Se jugará a todos los juegos y se realizará la rúbrica final sobre su proyecto realizado.

Sesión 21 Presentación de los videojuegos al alumnado de 3º ESO

- Descripción: Cada grupo presentará a los alumnos de 3º de la ESO el videojuego realizado con Scratch
- Actividades: a través de la pizarra digital cada grupo presentará su trabajo.
- Posteriormente los alumnos/as de 3º ESO jugarán a los diferentes videojuegos realizados con los ordenadores.

4. EVALUACIÓN Y RÚBRICA.

4.1 Evaluación

-La evaluación final se realizará como combinación de:

- 20% Entregas (individuales y en grupo)

Deben realizarse a tiempo al menos el 80% de todas las entregas. En caso contrario la calificación de la asignatura será No Presentado.

La nota de esta componente se reducirá en función del número de entregas realizadas en la fecha límite prevista)

- 20% El trabajo individual y sus problemas a entregar durante las sesiones previas al proyecto final. Las entregas son tres: expresión artística, historia y videojuego con Scratch.
- 40% El proyecto final de grupo

La calificación del proyecto tiene tres elementos:

-Primer prototipo (10%): Nota de grupo

-Segundo prototipo (20%): Nota de grupo (evaluación por otros grupos)

-Ampliación individual (10%): Nota individual:

0 si la ampliación del proyecto NO está bien

5 si la aplicación del proyecto está bien

10 si todos los miembros del grupo hacen bien la ampliación

- 10 % Autoevaluación de cada grupo mediante la rúbrica con Scratch
- 10 % Calificación subjetiva individual

Permite valorar participación, actitud, liderazgo dentro del grupo, etc.

4.2 Rúbricas

Las Rúbricas o Matrices de Valoración son herramientas valiosas para hacer una evaluación integral y auténtica cuando se realizan proyectos de clase en los que se utilice el entorno de programación de computadores Scratch. Éstas no sólo ayudan a guiar/direccionar el trabajo del estudiante (proceso) sino a valorar la calidad de los productos que han ejecutado (resultado).

Esperamos que esta rúbrica sirva a los docentes no sólo para establecer los criterios que los estudiantes deben seguir cuando trabajan con Scratch, sino como pauta que les permita a estos comparar sus propios avances en la realización de los proyectos de programación.

ASPECTOS	%	Excelente	Bien	Regular	Necesita mejoras
		5	4	3	1
Proceso	10%	<p>__ Diligencia correctamente los 5 elementos que componen la plantilla de análisis de problemas:</p> <p>1) Formular el problema;</p> <p>2) Resultados esperados;</p> <p>3) Datos disponibles;</p> <p>4) Restricciones;</p> <p>5) procesos necesarios[1]</p> <p>__ Utiliza productivamente el tiempo asignado para realizar el proyecto. Lo culmina antes del plazo de entrega estipulado.</p> <p>__ Colabora con sus compañeros, incluso, fuera del tiempo de clase.</p>	<p>__ Diligencia correctamente 4 de los 5 elementos que componen la plantilla de análisis de problemas:</p> <p>1) Formular el problema;</p> <p>2) Resultados esperados;</p> <p>3) Datos disponibles;</p> <p>4) Restricciones;</p> <p>5) procesos necesarios [1]</p> <p>__ Utiliza productivamente el tiempo asignado para realizar el proyecto. Cumple con el plazo de entrega de este.</p> <p>__ Colabora adecuadamente con sus compañeros de clase.</p>	<p>__ Diligencia correctamente 3 de los 5 elementos que comprende la plantilla de análisis de problemas:</p> <p>1) Formular el problema;</p> <p>2) Resultados esperados;</p> <p>3) Datos disponibles;</p> <p>4) Restricciones;</p> <p>5) procesos necesarios [1]</p> <p>__ La mayoría del tiempo de clase lo utiliza para realizar el proyecto. Cumple con dificultad el plazo de entrega.</p> <p>__ Colabora con sus compañeros de clase en pocas ocasiones.</p>	<p>__ Diligencia correctamente menos de 3 de los 5 elementos que comprende la plantilla de análisis de problemas:</p> <p>1) Formular el problema;</p> <p>2) Resultados esperados;</p> <p>3) Datos disponibles;</p> <p>4) Restricciones;</p> <p>5) procesos necesarios [1]</p> <p>__ No utiliza productivamente el tiempo asignado para realizar el proyecto. No cumple con el plazo de entrega.</p> <p>__ No colabora con sus compañeros de clase.</p>
Funcionamiento	10%	El programa realizado está completo (cumple con lo planteado por el docente en el proyecto de clase) y funciona correctamente.	El programa realizado no está completo (no cumple con lo planteado por el docente en el proyecto de clase), pero funciona correctamente.	El programa realizado no está completo (no cumple con lo planteado por el docente en el proyecto de clase) y funciona parcialmente.	El programa realizado no está completo (no cumple con lo planteado por el docente en el proyecto de clase) y no funciona.

Interfaz Gráfica	10%	<input type="checkbox"/> El programa realizado está organizado, tiene varios niveles y su diseño es complejo. <input type="checkbox"/> La interfaz gráfica es clara, tiene estructura y se adapta tanto al contenido como al diseño del programa. <input type="checkbox"/> Es fácil interactuar con el programa.	<input type="checkbox"/> El programa realizado está organizado, tiene dos niveles y su diseño es medianamente complejo. <input type="checkbox"/> La interfaz gráfica es clara pero tiene poca relación con el contenido y con el diseño del programa. <input type="checkbox"/> Es fácil interactuar con el programa.	<input type="checkbox"/> El programa realizado está poco organizado, tiene un solo nivel y su diseño es simple/sencillo. <input type="checkbox"/> La interfaz gráfica es poco clara y tiene escasa relación tanto con el contenido como con el diseño del programa. <input type="checkbox"/> Es difícil interactuar con el programa.	<input type="checkbox"/> El programa realizado no está organizado y su diseño es básico. <input type="checkbox"/> La interfaz gráfica es confusa. <input type="checkbox"/> No permite que otras personas puedan interactuar con el programa.
Creatividad	10%	El programa realizado es muy original y evidencia un grado de creatividad excepcional por parte del estudiante.	El programa realizado es original y refleja la creatividad 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 se basa totalmente en el diseño e ideas de otros. No se evidencia ninguna creatividad por parte del estudiante.

Autoevaluación de cada grupo:

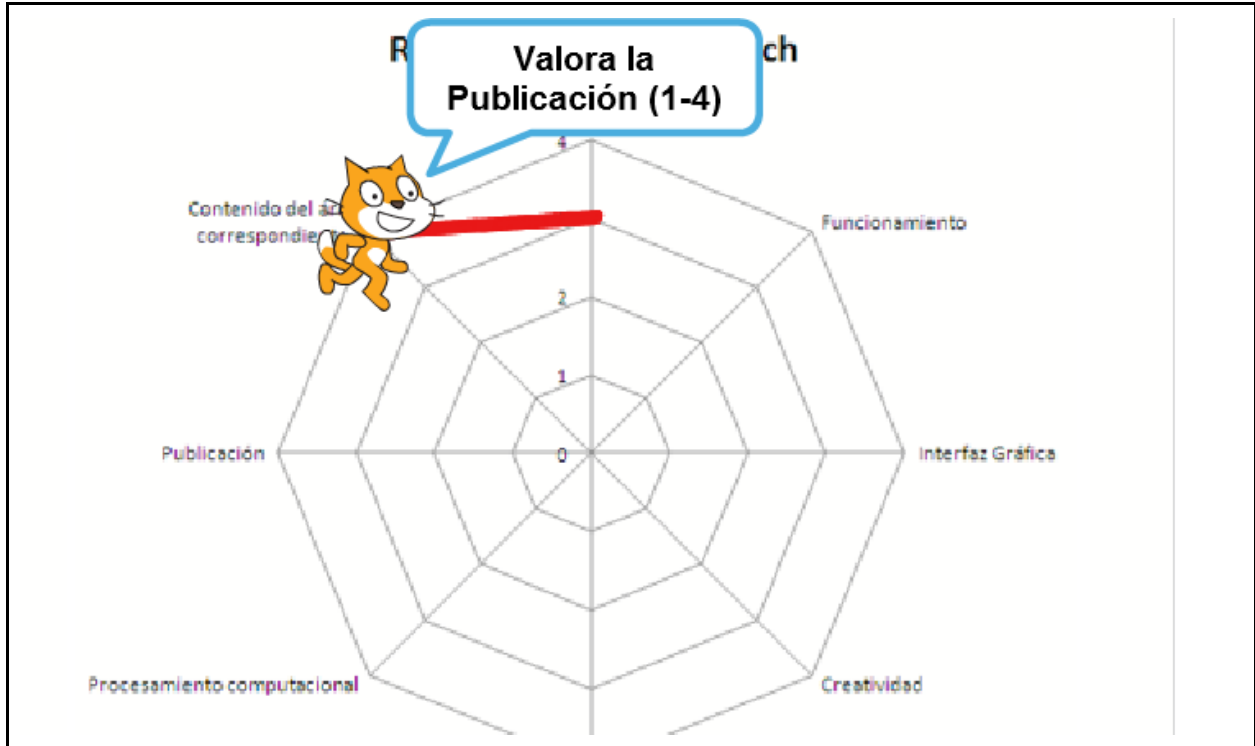
El proyecto “Rúbrica Scratch” es un modelo de programa que permite realizar una gráfica de la autoevaluación del proyecto realizado por un grupo de trabajo en el aula.

Enlace del Juego Scratch que permite evaluar entre los propios alumnos los diferentes juegos realizados en el aula: <http://scratch.mit.edu/projects/19302353/>

Lo primero que pide el programa al usuario es el nombre del grupo (puede ser el número de grupo). El nombre servirá para identificar la gráfica.

Posteriormente el programa pide que se introduzcan valores del 1 al 4 para puntuar las diferentes dimensiones que se evaluarán. Cada vez que introducimos un valor el gato se va desplazando y va dibujando una línea sobre un fondo de ejes tipo radial o tela de araña.

Realizada la gráfica, se pedirá a los alumnos que capturen la pantalla e incluyan la imagen en el informe del proyecto a entregar o bien se lo envíe al docente a través de un correo electrónico o plataforma de aprendizaje correspondiente.



ABP 2

S4A + ARDUINO

EFICIENCIA ENERGÉTICA EN EL CENTRO EDUCATIVO



6.2 PBL con S4A+Arduino. Eficiencia energética en el centro educativo

1. INTRODUCCIÓN:

Este trabajo pretende aplicar la metodología ABP para el desarrollo de contenidos del currículum de la asignatura de Tecnología en 4º ESO Enseñanza Secundaria Obligatoria.

Se ha diseñado un 2º proyecto como continuación del proyecto anterior con Scratch. El conocimiento adquirido en programación por bloques con Scratch en el anterior proyecto nos abre las puertas a poder utilizar la plataforma libre de Arduino ya que se rompe la barrera de su programación gracias a S4A (herramienta de programación de aplicación para Arduino y similar a Scratch).

Con este proyecto se trata por un lado que el alumnado cree una aplicación más o menos real con Arduino (basándose en su programación con S4A) y por otro lado que se conciencien sobre que el calentamiento global es una realidad, **pero también está en nuestras manos el llevar a cabo acciones que lo minimicen.**

Para ello se ha diseñado un ABP en el cual el alumnado continúa con la temática del calentamiento global. Con la perspectiva de fondo del plan 2020 “Ahorro de energía en Europa”, se pretende con esta actividad que el alumnado colabore en el aumento de la eficiencia energética del centro educativo.

Esta mejora consistirá en el estudio y planteamiento de una **mejora en el ahorro energético de la iluminación del centro educativo.** Son muchas las aplicaciones desarrolladas a través de Arduino en cuanto a este tema de eficiencia, domótica, etc. Debido a su idoneidad, los alumnos desarrollarán un sistema donde una vez estudiados el funcionamiento y consumo eléctrico de la iluminación en diversos puntos del centro, planteen una solución de mejora en cuanto a eficiencia energética en su iluminación. Ya sea mediante la automatización de la iluminación a través de sensores, cambio de luminaria más eficiente, tiempos encendido/apagado, etc.

1.1 Contexto

El proyecto está pensado para un grupo de alumnos de 4º ESO y se pretende cubrir el bloque 3 y 7 del currículum de Tecnológica y el bloque 5 del currículum de Ciencias Naturales, según el Decreto Foral 25/2007 para estas dos asignaturas.

Tecnología:

Bloque 3. Electrónica.

- Electrónica analógica. Componentes básicos, simbología, análisis y montaje de circuitos elementales.
- Electrónica digital. Aplicación del álgebra de Boole a problemas tecnológicos básicos. Puerta lógicas.
- Uso de simuladores para analizar el comportamiento de los circuitos electrónicos.

Bloque 7. Tecnología y sociedad.

- Aprovechamiento de materias primas y recursos naturales.
- Adquisición de hábitos que potencien el desarrollo sostenible.

1.2 Metodología

La metodología es la de ABP y se va a utilizar en la asignatura de Tecnología.

Se basa en los siguientes puntos:

- El aprendizaje se construye en base a los conocimientos previos que tienen los alumnos.
- Los alumnos aprenden trabajando e investigando en grupo.
- El profesor realiza el papel de tutor acompañando a los alumnos, su papel está muy lejos del profesor magistral.
- Se aprende en base a problemas cercanos al alumno y que cubre el currículo de su formación.
- El contexto del aprendizaje es esencial en la formación del alumno.

Dentro de clase se presenta el problema que los estudiantes deben resolver, se discuten las reglas, las probables soluciones, el tiempo máximo del proyecto, limitación de materiales, cómo se debe presentar la solución del problema.

Durante la ejecución del proyecto el profesor se convierte en un facilitador, es decir, a solicitud del estudiante actúa como consejero, para ayudar que el estudiante descubra e investigue por sí mismo. Permitiendo que el estudiante actúe con la máxima libertad posible.

Cada grupo presenta su proyecto de acuerdo a las bases de cada proyecto, se revisa la documentación y se realizan la prueba.

1.3 Objetivos

Los objetivos con esta metodología son:

- Inculcar la organización y la importancia del trabajo en equipo y potenciar actitudes flexibles y responsables, en la toma de decisiones, ejecución de tareas y búsqueda de soluciones basándose en la cooperación, imaginación y creatividad.
- Abordar con autonomía y creatividad problemas tecnológicos sencillos trabajando de forma ordenada y metódica (diseñar y construir objetos o sistemas que resuelvan el problema estudiado, y evaluar su idoneidad), basándose en el método de indagación.
- Desarrollar interés y curiosidad hacia la actividad tecnológica, generando iniciativas de investigación, así como de búsqueda y elaboración de nuevas realizaciones tecnológicas.
- Estimular la capacidad personal de aprendizaje, basándose en los aspectos más importantes del aprender a aprender y promover la responsabilidad por el aprendizaje propio.
- Aumentar el uso de las TIC para solucionar problemas tecnológicos.

- Amenizar la asignatura impartiendo clases más dinámicas y participativas, llevando a la práctica los contenidos teóricos y relacionándolos con la realidad.

A través de este proyecto se pretende además que los alumnos adquieran los conocimientos relacionados con los diferentes bloques de contenidos a trabajar:

Tecnología:

- Insistir en la capacidad personal de aprendizaje, desarrollando los aspectos más importantes del aprender a aprender.
- Que sean más ordenados.
- Promover los experimentos, donde el equivocarse es parte del aprendizaje y el autodescubrimiento.
- Ser más responsables con sus cosas.
- Desarrollar mayor movilidad en sus manos.
- Desarrollar sus conocimientos.
- Desarrollar la habilidad en grupo, permitiendo a las personas socializar.
- Desarrollar sus capacidades creativas.
- Poder observar cada detalle.
- Desarrollar el aprendizaje en forma divertida.
- Trabajar la motivación de los alumnos como motor central de su aprendizaje.
- Potenciar el desarrollo de las competencias que serán necesarias en el mundo futuro de los alumnos.
- Provocar en los alumnos los conflictos necesarios para estimular su aprendizaje.

1.4 Contribución a la adquisición de competencias básicas

El trabajar con metodología ABP está íntimamente relacionado con el trabajo por competencias, además de con la herramienta Scratch que impulsa y refuerza la utilización de muchas de ellas. Se van a enumerar las ocho competencias básicas, las cuales se van a poder trabajarse en este ABP.

Conocimiento e interacción con el mundo físico.

–Realiza pequeñas investigaciones sobre diferentes fenómenos reflejando interpretaciones cualitativas en un informe sencillo. Diseña, planifica y monta aparatos formados por elementos simples

La competencia matemática está íntimamente asociada

–Maneja con fluidez los números resolviendo y comprobando la validez del resultado en problemas de su vida real.

–Identifica y describe regularidades y tiene destreza para resolver problemas algebraicos sencillos.

– Reconoce figuras geométricas en su entorno directo, las clasifica según propiedades, estima y calcula longitudes, superficies y volúmenes para resolver retos geométricos en la elaboración de maquetas.

Tratamiento de la información y competencia digital

–Busca y selecciona información en distintas fuentes y soportes, la organiza y contrasta de forma guiada para realizar sencillas investigaciones que redacta aplicando modelos de elaboración de informes y trabajos de síntesis, valorando la importancia de su planificación.

–Utiliza Internet, redes interactivas y herramientas de comunicación con fines educativos o de relaciones sociales, mediante el intercambio de información, realización de trabajos cooperativos, uso de foros de dudas, de temas, discusión y envío de actividades y tareas.

Aprender a aprender

– Atribuye sus resultados a causas controlables y se muestra dispuesto a modificar sus estrategias para conseguir sus metas, identificando lo que sabe y lo que le queda por aprender.

–Se hace preguntas sobre los objetivos, pasos y procedimientos para planificar y realizar las diferentes tareas, revisándolas con pautas de autocorrección y comienza a pensar en alternativas de mejora.

Competencia social y ciudadana

–Establece relaciones de respeto en el trabajo cooperativo aceptando las responsabilidades propias del rol desempeñado

Comunicación lingüística

–Lee textos periodísticos, literarios y técnicos, redactando en distintos soportes mensajes con claridad para compartir sus ideas y conocimientos

–Selecciona adecuadamente sus palabras al expresar pensamientos y emociones y participa abiertamente en debates y exposiciones escolares explicando ordenadamente sus conocimientos sobre sucesos observados y temas conocidos

Autonomía e iniciativa personal

–Muestra un nivel aceptable de seguridad y confianza que se refuerza ante sus iguales, expresando sus ideas y aceptando las contribuciones de los otros en el desarrollo de su función en equipos cooperativos

Competencia cultural y artística

–Comunica mensajes con imaginación y creatividad, de forma individual o grupal, mediante producciones sencillas, eligiendo los materiales e instrumentos de uso escolar adecuados, incluidos los relacionados con las tecnologías de la información y comunicación.

En la secuenciación de las sesiones para realizar el proyecto se refleja que competencias se trabajan en cada sesión.

2. DESCRIPCION DEL PROYECTO

ESCUELAS EFICIENTES. EL CAMBIO CLIMÁTICO EN EL CENTRO EDUCATIVO

En relación con el proyecto anterior con Scratch, se va a realizar el segundo proyecto orientado a S4A+Arduino. Continuando con la temática del cambio climático vamos a aprovechar las posibilidades de la combinación de S4A y Arduino para mejorar la eficiencia energética de la escuela.

Se trata de contribuir al ahorro de energía mediante eficiencia lumínica en los pasillos, baños y aulas del centro. Para ello deberemos diseñar diferentes sistemas de dificultad similar que tengan un ahorro energético mejorando las instalaciones ya existentes.

- Pasillos: Cuando alguien se mueve en el pasillo, la iluminación se activa, lentamente, sube poco a poco, se mantiene unos 30 segundos y luego empieza a bajar lentamente hasta que se apaga, si algo se mueve mientras tanto, vuelve a subir y sigue con el ciclo regulando el encendido y el apagado, hasta que si no hay nadie, se apaga.
- Baños: Se realizará el mismo sistema que en los pasillos.
- Aulas. Un sensor de luz y un potenciómetro de ajuste por parte de los usuarios regularán el nivel de luz deseado en el aula. De esa manera el nivel de luz irá variando en función de la luz disponible.

Video iluminación eficiente

2.1 Duración

El proyecto durará 5 semanas con 15 sesiones.

De esas 15 sesiones 6 corresponden a la realización de una serie de ejemplos o prácticas con Arduino+S4A para familiarizarse con la herramienta y posteriormente el grupo realizará el proyecto final durante las siguientes 3 semanas.

2.2 Grupos

Se utilizará la técnica: “el puzzle”.

Aunque puede variar la tipología de la clase y la cantidad de alumnos que haya en él, se realizarán grupos de 3-5 personas, donde a cada grupo se le propondrá un mismo proyecto.

Los alumnos matriculados en estos módulos son 16, por lo que se harán 4 grupos de 4 alumnos. Estos grupos deben ser mixtos, incluir alumnos y alumnas, y estar equilibrados.

Después de la presentación del proyecto y la formación de grupos, se reunirán para elaborar el acta de constitución del grupo, donde deben indicar:

1. Nombre de los miembros que forman el grupo.
2. Franjas horarias fuera de clase en las que todos los miembros del grupo están libres para reuniones por si fuese necesario.
3. Acuerdo en cuanto al nivel de ambición del grupo (“todos vamos a por la máxima nota” o “todos vamos simplemente a por el aprobado”).
4. Miembro responsable en cada grupo.
5. Asignación de tareas dentro de cada grupo: búsqueda de información, programación, diseño del disfraz...

Si el grupo no encuentra franjas horarias libres en común o no están de acuerdo en el nivel de ambición entonces se producen los cambios de grupo necesarios.

Cada miembro del grupo se asigna unas tareas, que debe realizar con los recursos disponibles. Acabada la sesión de clase, cada miembro del grupo base va cumplimentando su parte de la memoria de lo trabajado. Después, cada grupo debe reunirse para que, por turnos, cada miembro explique en 5 minutos al resto su parte del trabajo de la sesión anterior.

A lo largo de las sesiones, en clase se realizará la reunión de responsables. Miembros de cada grupo se unen para compartir dudas y profundizar en el aprendizaje de su parte del trabajo.

El trabajo del proyecto de los grupos debe repartirse en tareas, que están desarrolladas en el apartado de secuenciación. Se va a estructurar esas tareas en dos tipos, una en cuanto a la organización del grupo y la otra de las tareas propias para el desarrollo del proyecto::

1- Organización del grupo: elección de responsable, descripción de las tareas del responsable.

Reparto de responsabilidades en el grupo:

- Coordinador: coordinación del grupo, comunicación con profesor, entrega y recogida del proyecto.
- Material y limpieza: recoger y guardar el material, coordinación de la limpieza.
- Herramientas: comprobación a la entrada y a la salida de la clase de que están todas las herramientas. Velar por su adecuado uso.
- Documentación: rellenar el cuaderno de control y desarrollo del proyecto del grupo. Rellenar la hoja de progreso diario.

2-El trabajo de cada grupo debe repartirse en tareas. Cada uno del grupo será el experto de un área dentro de la fabricación del montaje que permite tener una iluminación eco-suave.. Esas partes son:

- ➔ Diseño y planos del circuito eléctrico, sensores, resistencias, salidas a LED's
- ➔ Programación Algoritmo mediante programa S4A
- ➔ Implementación del montaje dentro de la instalación de la escuela. Construir caja, cableado alimentación, etc. parte hardware.
- ➔ Estudio lumínico y de ahorro energético. Que tenemos en el centro y como se puede mejorar. Cambio de tipos de luz por una mas eficientes pero sin perder intensidad lumínica, etc.

Al ser un trabajo colaborativa mediante la técnica puzzle los expertos de cada área se reunirán con los grupos de expertos de los demás grupos y sus deliberaciones las explicaran a los demás miembros de su propio grupo para que al final todos sepan de todo.

3. DESARROLLO DEL PROYECTO

3.1 Secuenciación

Sesión 1

-Introducción y repartición de la guía: explicación del método APB, trabajo grupal, cómo se va a evaluar, los contenidos que se van a ver, etc.

-Se hará una pequeña introducción de lo que es Arduino (enlaces red para ver algún vídeo o la página Web,...)

-Se utilizará un vídeo corto que tenga que ver con Arduino y su aplicación en domótica.

En esta sesión se ven un poco las aplicaciones de los sistemas automáticos de control.

Sesion 2 (puzzle)

El profesor, dividirá a la clase en grupos reducidos cuatro alumnos de forma heterogénea (los mismos que harán el proyecto). A estos se les facilitará el mismo número de textos en los que el tema quede estructurado. Así, cada experto tratará su área el trabajo y estudio de su parte a trabajar. Las áreas ya las hemos comentado; diseño circuito eléctrico, programación con S4A, Implementación del montaje dentro de la instalación de la escuela, estudio lumínico. Con un tiempo limitado y escaso para fomentar la concentración, cada uno de ellos deberá leer y aprender la parte asignada. Acto seguido se establecerá una reunión de “expertos” entre cada uno de los miembros de todos los grupos que han estudiado el mismo texto. Así la clase quedará formada por subgrupos hechos a partir de los grupos originales que debatirán sobre lo aprendido. Habrá un ponente que deberá exponer el contenido tal y como lo ha entendido, y un secretario que deberá asegurarse de que la discusión se realiza en el tiempo asignado. Una vez pasado ese breve tiempo, cada uno de los “expertos” en una de esas cuatro partes en las que se ha desmenuzado el tema, deberá elaborar individualmente un guión para explicar lo aprendido y se ha matizado en la puesta común sobre su texto. Entonces, se reúne con los demás miembros del grupo original que habrán hecho lo propio con su parte, y deberán esforzarse en explicarse unos a otros con la máxima solvencia lo aprendido.

Para evaluar lo que han aprendido los alumnos con el “puzzle”, el profesor realizará una pequeña ronda de preguntas a alguien del grupo al azar.

En esta sesión se descargará e instalará los softwares necesarios para funcionamiento de Arduino y S4A.

Sesión 3 (puzzle)

Se le dará seguida a la anterior sesión. El profesor recordará lo aprendido el día anterior y se seguirá con el mecanismo del puzzle.

Una vez que hayan acabado todos los participantes de los grupos y se hayan hecho las preguntas pertinentes se realizará un pequeño cuestionario para evaluar mejor lo aprendido mediante el puzzle.

Sesión 4, 5, 6 (Ejemplos Arduino y programación S4A)

Ver ejemplos tipo de Arduino+S4A para su comprensión.

Aunque durante la actividad anterior ya hemos visto suficiente contenido de programación con Scratch (en la cual se basa S4A) durante esta clase veremos la variantes de S4A respecto a Scratch y su aplicación en Arduino. Para ello se procederá en estas sesiones a introducir ejemplos significativos, los cuales pueden servir para desarrollar ideas para el proyecto así como para adquirir destrezas antes de pasar a una fase donde los grupos deben desarrollar sus propios productos.

Sesión 7 (proyecto)

Se les hace a los alumnos la pregunta motriz, y se les explica que han de realizar un proyecto de “iluminación eficiente” (una maqueta) que deberá estar provisto, además, de un documento o “Memoria” hecho en grupo que contenga al menos: Cuaderno del alumno, (el desarrollo del proyecto día a día), finalidad del circuito, esquema de entradas y salidas, diagrama de estados, esquema y lista de materiales, el programa con comentarios, comandos nuevos usados, diciendo para qué sirve cada comando o función nueva usado (Variables, constantes...). Por último, tendrán que hacer una “Presentación del proyecto” que no supere las 10 hojas, y lo tendrán que exponer delante de la clase en no más de 5-10 minutos.

Las siguientes sesiones irán dirigidas a la realización de la maqueta y la memoria del proyecto. Por lo que no se detallan dado que el ritmo de trabajo puede diferir. En todo caso el profesor irá evaluando a los grupos, resolviendo dudas y dirigiéndoles, echándoles un vistazo a todos los proyectos que se van realizando (en los ordenadores irán guardando el programa que van desarrollando, y las maquetas se guardarán en el almacén del aulal taller

Sesión 8, 9, 10, 11, 12, 13: (Proyecto).

Desarrollo y realización de la maqueta y la memoria. Profesor como guía, supervisor y resolución de dudas. Se velará por que los alumnos avancen de manera estipulada y no se queden rezagados, ayudándoles en ese caso a resolver sus problemas mediante pistas, etc.

Sesión 14 (presentación de los proyectos y autoevaluación sobre el trabajo realizado)

Primero se presentará oralmente el Proyecto (el profesor elige a cualquiera de los del grupo para que lo presente y después hará alguna pregunta al azar a cualquiera de ellos).

A través de rúbrica los grupos se realizarán una autoevaluación sobre su trabajo realizado.

Se pasarán unos cuestionarios que los alumnos tendrán que rellenar en grupo, consensuando la respuesta entre todos los miembros, en los que se hará una pregunta de reflexión sobre cómo ha funcionado el grupo y cómo debería funcionar en casos de dificultad.

Sesión 15 (Presentación a dirección del centro el estudio y mejora eficiencia realizada)

Los grupos presentarán sus aplicaciones al responsable del centro en la materia para poder analizar la puesta en marcha de las mismas para obtener un ahorro real de energía.

4. CRITERIOS DE EVALUACIÓN

4.1 Evaluación

Calificaciones

1. Requisitos mínimos	
Entregar los trabajos en las fechas programadas	
Todos los trabajos se entregarán cumpliendo el formato establecido (*)	
Obligatorio entregar al menos el 80 % de los trabajos en las fechas señaladas para poder aprobar	10%
Obligatorio entregar todos los trabajos del proyecto para presentarse al examen de la asignatura	
2. Proyecto	
Para obtener la nota máxima, todos los miembros del equipo deberán obtener la nota máxima	
Versión preliminar	10%
Versión final	25%
Ejercicio individual, exposición del proyecto	15%
3. Conocimientos básicos	
Necesario demostrar que se conocen todos los conocimientos básicos para aprobar	20%
Existen 3 convocatorias: trimestre, junio y septiembre	
4. Autoevaluación	10%
5. Iluminación eficiente (Autoevaluación)	10%

Rúbrica para autoevaluación

Nombre Alumno: _____

Saberes a alcanzar en la materia	Insuficiente (4- 5)	Suficiente (5 – 6)	Notable (7 – 8)	Sobresaliente (9 – 10)
Saber (33%)				
Conocer y comprender los contenidos de la materia				
Saber hacer (33%)				
Buscar información				
Analizar la información				
Tomar decisiones, resolver problemas				
Comunicarse				
Saber ser y convivir (33%)				
Trabajar en equipo: participa, coopera, se compromete,.....				
Es asertivo, esta motivado				
Tiene deseo de aprender, iniciativa, es autónomo, sabe regular el autoaprendizaje				
NOTA FINAL				

Rúbrica para calificar el proyecto

Niveles de consecución				
critérios	4	3	2	1
CONOCE Y RESUELVE	Usa una estrategia eficiente y efectiva para resolver	En general, usa una estrategia eficiente y efectiva para resolver	Algunas veces usa una estrategia efectiva para resolver	Raramente usa una estrategia efectiva para resolver
APLICA MÉTODOS DE SOLUCIÓN	En todo el proyecto el alumno demuestra el uso correcto de los métodos concordantes con los temas propuestos para la solución de ejercicios y problemas.	En gran parte del proyecto el alumno demuestra el uso correcto de los métodos concordantes con los temas propuestos para la solución de ejercicios y problemas.	En el proyecto el alumno demuestra poca asimilación del uso correcto de los métodos concordantes con los temas propuestos para la solución de ejercicios y problemas.	En el proyecto el alumno demuestra una deficiente asimilación del uso correcto de los métodos concordantes con los temas propuestos para la solución de ejercicios y problemas.
RESOLUCIÓN DE EJERCICIOS Y PROBLEMAS	Usa una estrategia eficiente y efectiva para resolver ejercicios y problemas que se presentan	Por lo general, usa una estrategia efectiva para resolver ejercicios y problemas que se presentan	Algunas veces usa una estrategia efectiva para resolver ejercicios y problemas que se presentan	Raramente usa una estrategia efectiva para resolver ejercicios y problemas que se presentan
PRODUCCIÓN DE TEXTOS	Las instrucciones que se describen en el anteproyecto y manual son objetivas, precisas, detalladas y respetan el orden secuencial.	Las instrucciones que se describen en el anteproyecto y manual son detalladas y respetan el orden secuencial, pero carecen de objetividad y precisión.	Las instrucciones que se describen en el anteproyecto y manual son objetivas, precisas y detalladas, pero carecen de orden secuencial	Las instrucciones que se describen en el anteproyecto y manual no son objetivas, precisas, detalladas ni respetan el orden secuencial
ESPECIFICACIONES	El resultado final se ajusta a todo lo especificado	El resultado final se ajusta en casi todo lo especificado	El resultado final se ajusta en parte a lo especificado	El resultado final no se ajusta a lo especificado
MEJORAS	Introduce varias mejoras	Introduce dos mejoras	Introduce una mejora	no introduce ninguna mejora

Rúbrica exposición del proyecto

Niveles de consecución				
critérios	5	4	3	1
Tono de voz	El alumno siempre modula correcta y apropiadamente el tono de voz. Siempre la comunicación oral fluye con naturalidad y corrección. Siempre utiliza el vocabulario correcto y adecuado	El alumno casi siempre modula correcta y apropiadamente el tono de voz. Casi siempre la comunicación oral fluye con naturalidad y corrección. Casi siempre utiliza el vocabulario correcto y adecuado	El alumno casi nunca modula correcta y apropiadamente el tono de voz. Casi nunca la comunicación oral fluye con naturalidad y corrección. Casi nunca utiliza el vocabulario correcto y adecuado	El alumno nunca modula correcta y apropiadamente el tono de voz. Nunca la comunicación oral fluye con naturalidad y corrección. Nunca utiliza el vocabulario correcto y adecuado
Comprensión	El alumno puede con precisión contestar casi todas las preguntas planteadas sobre el tema por sus compañeros de clase	El alumno puede con precisión contestar la mayoría de las preguntas planteadas sobre el tema por sus compañeros de clase	El alumno puede con precisión contestar unas pocas de las preguntas planteadas sobre el tema por sus compañeros de clase	El alumno no puede contestar las preguntas planteadas sobre el tema por sus compañeros de clase
Calidad presentación	El alumno mantiene siempre la atención de los espectadores. Evita leer lo que está escrito en su presentación	El alumno casi siempre mantiene la atención en los espectadores. Evita casi siempre leer lo que está escrito en su presentación	El alumno casi nunca mantiene la atención en los espectadores. Casi siempre lee lo que está escrito en su presentación	El alumno nunca mantiene la atención en los espectadores. Y siempre lee lo que está escrito en su presentación
Dominio del contenido	El alumno siempre demuestra dominio del contenido del tema	El alumno casi siempre demuestra dominio del contenido del tema	El alumno casi nunca demuestra dominio del contenido del tema	El alumno nunca demuestra dominio del contenido del tema

5. EVALUACIÓN DEL ÉXITO DE LA PROPUESTA DEL PBL

Para finalizar la actividad, me parece importante conocer el grado de aceptación de los alumnos ante este cambio metodológico en la manera de instruir parte de la materia de los dos módulos en los cuales se va a realizar el PBL.

Se trata de saber si esta metodología ha gustado al alumnado y se ha sentido sobre todo **más motivada** para afrontar las competencias. Como he comentado al inicio el aumento de su motivación era principalmente el objetivo inicial, además de trabajar los contenidos desde una realidad más práctica y colaborativa. Por eso para terminar el PBL se pasará un cuestionario al alumnado para conocer la aceptación del proyecto y su grado de motivación.

Para medir su grado de motivación durante el transcurso del proyecto el profesor intentará detectar la motivación del alumnado de la siguiente manera:

- La dirección de la atención y sus conductas de elección
- Persistencia y esfuerzo mostrado en la tarea
- Persistencia y motivación continuada
- El logro del objetivo marcado y su rapidez y calidad en la consecución de este objetivo
- Finalmente una serie de cuestionarios específicos que miden el nivel de motivación.

7. BIBLIOGRAFÍA:

- [1] Fracaso y abandono escolar en España, Mariano Fernández Enguita, Luis Mena Martínez y Jaime Riviere Gómez.
- [2] Programación de computadores y desarrollo de habilidades de pensamiento en niños escolares: fase exploratoria, Hernando Taborda Diego Medina, Universidad ICESI Cali, Julio, 2012.
- [3] Aprendizaje basado en problemas, ABP, Universidad Veracruzana.
- [4] Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Siver, J., Silverman, B., y Kafay, Y. (2009). Scratch: Programming for all. *Communications of the ACM*, 52 (1), 60-67.
- [5] Radenski, A. (2006). "Python First": A lab-based digital introduction to computer Science. ITiCSE '06 11th Annual Conference on Innovation and Technology in Computer Science Education Bologna, Italy -June 26 - 28, 2006.
- [6] Blank, 1997; Dickinson, et al, 1998; Harwell, 1997
- [7] Personalización del proceso de enseñanza-aprendizaje, Adrián Zúñiga Antón.
- [8] Esteban, 2008a, 2009b; Esteban & Branda, 2008; Molina, García, Pedraz & Antón, 2003; Planella, Escoda & Suñol, 2009.
- [9] Vigotsky y Krahlen, zona de desarrollo próximo.
- [10] APRENDIZAJE BASADO EN PROBLEMAS PROBLEM BASED LEARNING, Patricia Morales Bueno y Victoria Landa Fitzgerald
- [11] Esteban, 2009a; Esteban & Branda, 2008.
- [12] Aprendizaje Basado en Proyectos (PBL) en las enseñanzas técnicas. Miguel Valero-García Febrero 2008.
- [13] La visión de Seymour Papert para la educación de la niñez, Catherine Wilson Gillespie, Escuela de Educación de Drake University.

8. ANEXOS

ANEXO 1

Material relacionado con ABP 1 con herramienta Scratch

Enlaces para obtener material para realizar los ABP de Scratch:

<http://scratched.media.mit.edu/sites/default/files/CurriculumGuide-v20110923.pdf>

http://learnscratch.org/learnscratch/LP_Scratch1.pdf

Manuales de Scratch:

<http://download.scratch.mit.edu/ScratchReferenceGuide14.pdf>

<http://nuestratecnologia.files.wordpress.com/2011/09/manual-scratch.pdf>

Cuaderno de trabajo con Scratch:

<https://drive.google.com/?tab=mo&authuser=0#folders/0B8cEpb3NnO8WOWpjYnNheUtiOW>

[M](#)

ANEXO 2

Material relacionado con ABP 2 con herramientas S4A+Arduino

Enlace material para realización de prácticas con S4A + Arduino

<http://vps34736.ovh.net/S4A/s4a-manual.pdf>

<http://recursostic.educacion.es/observatorio/web/es/software/software-educativo/1018-monograficodesarrollos-de-scratch-para-robotica-enchanting-y-s4a?start=3>