

Algoritmos ecológicos

Escuela: ESEA en Danzas N°1 DE 20 “Profesora Nelly Ramicone”

Autora: Patricia Colombarini

Sala/grado/año: Tercer año

Breve descripción

El objetivo central de esta práctica es que los estudiantes apliquen activamente sus conocimientos de algoritmos, pseudocódigo y estructuras de programación mediante la creación de un programa funcional, utilizando la plataforma Scratch. Se desafió a los alumnos a diseñar y codificar un juego de preguntas y respuestas en formato Verdadero o Falso.

Este proyecto de cierre de unidad requirió que los estudiantes pusieran en práctica de forma integral los temas vistos, desde la planificación lógica del algoritmo hasta la implementación de estructuras condicionales en el entorno de Scratch.

Los alumnos reutilizan su código y lo adaptan para crear un segundo juego V/F enfocado en el cuidado del medio ambiente. Esta adaptación servirá como cierre de un proyecto anual más amplio.

Esta metodología fomenta la reusabilidad del código y demuestra que las habilidades de programación son herramientas transferibles y aplicables a cualquier campo de estudio. En esencia, la propuesta transforma la teoría en una herramienta de aprendizaje práctica, lúdica y con impacto temático.

Situación inicial

La práctica pedagógica surge de la necesidad de superar un desafío común en la enseñanza de Programación y Computación, el riesgo de abstracción teórica.

Se observa que, a pesar de comprender los conceptos de algoritmo, pseudocódigo y estructura de control a nivel teórico o práctica guiada, muchos estudiantes enfrentan dificultades al trasladar estos conocimientos a una

aplicación práctica y funcional. El conocimiento tendía a permanecer en el plano abstracto, desconectado de su potencial de creación.

La actividad no solo evalúa la memorización, sino la aplicación integral y la síntesis de todos los temas vistos (planificación lógica, diseño algorítmico, y codificación estructurada).

El diseño de la práctica se fundamentó en dos pilares clave, buscando una transición efectiva de la teoría a la práctica y la promoción de habilidades transferibles:

SCRATCH Y EL FORMATO "JUEGO V/F"

Se eligió la plataforma Scratch por su entorno visual, lúdico e intuitivo, que baja la barrera de entrada a la codificación, permitiendo a los estudiantes enfocarse en la lógica algorítmica y en la implementación de estructuras condicionales (la clave para un juego de preguntas y respuestas) sin frustrarse con la sintaxis compleja de otros lenguajes. El juego de V/F es el vehículo ideal para aplicar estructuras de decisión (Si/Entonces/Si no) de manera clara y funcional.

LA REUSABILIDAD DEL CÓDIGO (DOBLE PROYECTO)

La principal motivación detrás de la doble entrega (el juego inicial y su adaptación al tema del cuidado del medio ambiente) es demostrar la transferibilidad del pensamiento computacional. Al desafiar a los alumnos a reutilizar y adaptar su código base, se refuerza un principio fundamental de la ingeniería de software y se les muestra que la programación es una herramienta transversal: una vez que dominan la lógica de un programa, puede aplicarse a cualquier campo de estudio o tema de impacto (en este caso, la conciencia ambiental).

Se promueve la eficiencia y la abstracción sobre la creación desde cero.

La experiencia se diseñó para transformar la teoría de algoritmos en una herramienta práctica, lúdica y transferible, asegurando que el conocimiento sea una capacidad de creación y no solo de comprensión.

Objetivos

El objetivo es que el conocimiento no sea pasivo, sino que se demuestre en un producto funcional y ejecutable (el juego V/F en Scratch).

También generar una experiencia de aprendizaje que promueva el principio de código reutilizable. Se busca intencionalmente que los alumnos valoren la

eficiencia del diseño algorítmico y perciban la programación como un conjunto de herramientas adaptables.

Utilizar un proyecto de programación como medio para abordar y darle sentido temático a un contenido curricular más amplio. Se demuestra que la habilidad de programar es una herramienta de impacto y expresión aplicable a cualquier campo de estudio.

Para lograr la intencionalidad descrita, la práctica creó las siguientes condiciones:

- * Condición de desafío integrador.
- * Planificación lógica (algoritmo/pseudocódigo).
- * Ejecución técnica (implementación en Scratch).
- * Condición de aprendizaje lúdico.
- * Condición de transferencia explícita.

Se espera que, al finalizar esta práctica, los y las estudiantes sean capaces de:

- * **APLICAR ESTRUCTURAS LÓGICAS DE PROGRAMACIÓN:** aplicar de forma efectiva y funcional los conocimientos de estructuras condicionales (tales como Si/Entonces/Si no) y algoritmos en la creación de un programa interactivo.
- * **TRADUCIR LA PLANIFICACIÓN ALGORÍTMICA:** lograr un diseño lógico previo a un código funcional dentro del entorno de programación visual Scratch.
- * **DESARROLLAR LA CAPACIDAD DE REUTILIZACIÓN DE CÓDIGO:** demostrar la capacidad de reutilizar, adaptar y modificar un código base existente para generar un nuevo proyecto con una temática diferente, promoviendo la eficiencia en la codificación.
- * **INTEGRAR EL PENSAMIENTO COMPUTACIONAL CON CONTENIDOS TRANSVERSALES:** utilizar la programación como una herramienta para la creación de contenido y la expresión temática, vinculando las habilidades de codificación con temas de impacto social y ambiental (cuidado del medio ambiente).
- * **EVALUAR Y DEPURAR EL FUNCIONAMIENTO DEL CÓDIGO:** Poner a prueba la funcionalidad del juego de preguntas y respuestas para identificar y corregir errores en el flujo lógico y en la implementación de las estructuras condicionales.

Contenidos

Fundamentos del pensamiento computacional y algorítmico.

Pseudocódigo y algoritmo: diseño y representación de la secuencia lógica de instrucciones para un proceso (el juego V/F).

Los datos, el proceso y los resultados: identificación de los datos de entrada (respuestas del usuario) y variables (puntaje) que deben ser procesados por el software para generar un resultado (acierto/error, puntaje final).

Metodología de resolución de problemas computacionales: aplicación integral del ciclo completo de desarrollo: Análisis del problema (comprensión del juego), Identificación de los datos, diseño y representación del algoritmo.

Codificación, ejecución, prueba, depuración: implementación del algoritmo en el entorno Scratch.

Validación del software y depuración colaborativa para corregir fallos lógicos y de ejecución.

Estructuras de programación y control: uso primordial de las estructuras condicionales para validar las respuestas, variables V/F y dirigir el flujo del juego.

Implementación de variables para almacenar y gestionar el puntaje, el estado del juego o los contadores.

Utilización de eventos y mensajes (propios de Scratch) para coordinar las acciones y las transiciones entre preguntas o escenas del juego.

Destinatarios

Alumnos de tercer año de nivel secundario.

Secuencia didáctica

PRIMERA ETAPA: INTRODUCCIÓN Y ANÁLISIS

DOCENTE:

Presentación del Desafío: crear un juego V/F en Scratch.

Articulación de contenidos: repasar el concepto de problema computacional y las estructuras condicionales, bucles, variables, enfatizando su rol en la experiencia.

Introducir la metodología de resolución de problemas computacionales (análisis, diseño, codificación, prueba).

ALUMNOS:

En grupos (3-5 alumnos), analizan el problema. ¿Cómo se gestionan las entradas (respuesta del usuario) y salidas (acierto/error, puntaje) en un sistema digital?

Análisis del contenido (Biología). Eligen la temática de las preguntas y elaboran un banco inicial de preguntas V/F sobre Medio Ambiente.

Inician el diseño a través de pseudocódigo, definiendo la secuencia lógica antes del entorno de programación.

Roles y compromiso: definen roles dentro del grupo y establecer acuerdos de colaboración.

SEGUNDA ETAPA: CODIFICACIÓN INICIAL Y ESTRUCTURACIÓN

DOCENTE:

- Guía a los grupos en la correcta codificación de las estructuras en Scratch. Hace énfasis en el uso de variables para llevar la cuenta del puntaje.
- Demostración práctica: repaso de contenidos sobre cómo anidar condicionales o cómo usar los bloques.
- Supervisión de vocabulario: incentiva a los alumnos que utilicen el vocabulario informático de forma precisa al discutir su código.

ALUMNOS:

- Implementación del código: crear el algoritmo por medio de bloques de código en Scratch. Crear las variables necesarias (por ejemplo: puntaje, preguntas).
- Estrategias de programación: implementar las estructuras de programación para validar las respuestas V/F.
- Comunicación constante entre los integrantes del equipo por la división de tareas.

TERCERA ETAPA: EJECUCIÓN, PRUEBA Y DEPURACIÓN

DOCENTE:

- Orienta a los grupos en la fase de ejecución y prueba, indicando que deben probar todas las rutas lógicas (respuesta correcta, respuesta incorrecta) y casos extremos.
- Intervención en errores: sugiere modificaciones con preguntas guía (“¿Qué valor tiene esa variable en este punto?”, “¿El condicional está evaluando la condición correcta?”) para que los alumnos autónomamente encuentren el error.

ALUMNOS:

- Realizan pruebas exhaustivas del juego. Identifican los problemas computacionales (errores) que surgen y trabajan de forma colaborativa para depurar el código y asegurar su correcto funcionamiento.
- Presentan el primer juego V/F funcional. Reciben feedback de otros grupos y del docente sobre la usabilidad y la eficiencia del código.

CUARTA ETAPA: REUTILIZACIÓN

DOCENTE:

- Introduce el desafío de la reutilización del código para el segundo juego sobre cuidado del medio ambiente.
- Se pone en contacto con la docente de Biología para revisar el banco de preguntas V/F para asegurar su rigor científico y la relevancia de las pautas de cuidado y responsabilidad en el uso de la información.
- Evalúa la capacidad de modularización y eficiencia demostrada en la adaptación, así como la calidad del producto final y la presentación.

ALUMNOS:

- Realizan un análisis crítico del código base para identificar las secciones que representan la lógica (estructuras condicionales, variables de puntaje).
- Reutilizan los bloques de código que gestionan la secuencia y el puntaje. Solo se modifican los datos de entrada (las preguntas y las respuestas) del juego.
- Finalizan el segundo juego, demuestran que es funcional y exponen a la clase el proceso y las decisiones tomadas, utilizando el vocabulario informático preciso.

Otros actores

La experiencia contó con la participación de la docente de Biología, que actuó como actor clave y promotor temático. La práctica se realizó como cierre de su proyecto anual sobre el cuidado del medio ambiente. Esto aseguró la transversalidad de los contenidos, donde la programación (el cómo) sirvió como herramienta práctica para abordar el contenido disciplinar de Biología (el qué).

Evaluación**CRITERIOS DE EVALUACIÓN (¿QUÉ SE PROPUSO EVALUAR?)**

La evaluación fue un proceso continuo y formativo, culminando en la valoración del producto final y la presentación grupal, enfocada en la aplicación del pensamiento computacional y la colaboración.

Los criterios se enfocaron en la aplicación efectiva de la metodología y la transferencia de habilidades, alineados con los objetivos del cuarto bimestre:

* **APLICACIÓN DE ESTRUCTURAS CONDICIONALES:** correcta implementación de las estructuras de programación, eventos, control, variables para que el juego funcione lógicamente y el puntaje se asigne correctamente.

* **INTEGRACIÓN METODOLÓGICA:** completa aplicación del ciclo de resolución de problemas computacionales (desde el análisis y diseño algorítmico hasta la depuración).

* **EFICIENCIA Y REUTILIZACIÓN DEL CÓDIGO:** capacidad de los grupos para reutilizar y adaptar el código base del primer juego al segundo, demostrando una comprensión de la lógica de programación modular/por objetos.

- * COLABORACIÓN Y COMPROMISO GRUPAL: grado de participación equitativa y el compromiso de cada miembro con los roles definidos, evidenciado en la gestión del tiempo y la calidad de la integración de las partes.
- * RIGOR TEMÁTICO Y CONTEXTUALIZACIÓN: calidad de las preguntas elaboradas para el juego del medio ambiente y su alineación con las pautas de cuidado y responsabilidad curricular.
- * USO DE VOCABULARIO INFORMÁTICO: fluidez y precisión en el uso del vocabulario informático (ejemplos: variable, algoritmo, condicional) durante la presentación y la justificación de las decisiones de codificación.

INSTRUMENTOS DE EVALUACIÓN (¿CÓMO SE EVALUÓ?)

Se utilizaron distintos instrumentos para capturar la progresión y el resultado final del aprendizaje:

- * REGISTRO DE OBSERVACIÓN DOCENTE (FASE 1 Y 2): evaluación formativa continua. Observación de la participación individual y grupal durante la elaboración del pseudocódigo y la codificación inicial, evaluando la aplicación de la metodología de resolución.
- * CORRECCIÓN DEL ALGORITMO/PSEUDOCÓDIGO INICIAL: evaluación de proceso y revisión del diseño lógico previo a la codificación, asegurando que el análisis del problema y la secuencia de pasos sean correctos antes de pasar al entorno Scratch.
- * PRODUCTO FINAL (JUEGO V/F Y SU ADAPTACIÓN): evaluación sumativa del resultado, aplicada al código final en Scratch. Se evalúa la funcionalidad, la correcta implementación de las estructuras, las variables y la eficiencia en la reutilización del código.