

Explorando software libre y Open Source: ¡Descubre el código que mueve el mundo!

Tecnología e Informática | Pensamiento Computacional | Diseño Universal para el Aprendizaje

Descripción

En esta sesión, los estudiantes de media explorarán las diferencias y similitudes entre el software libre y el software Open Source, dos conceptos fundamentales en el mundo de la tecnología actual. Aprenderán qué significan estos términos, cómo influyen en el desarrollo de programas y por qué son relevantes para su vida diaria, ya que muchas aplicaciones y herramientas que usan cotidianamente están basadas en estos modelos.

El propósito es que los jóvenes comprendan la importancia del acceso abierto al conocimiento y al código fuente, fomentando una visión crítica y ética sobre el uso y desarrollo del software. Además, se estimulará el pensamiento computacional al analizar casos reales y participar activamente en el aprendizaje, promoviendo competencias digitales esenciales para su futuro académico y profesional.

Esta experiencia se conecta con su realidad al mostrar ejemplos concretos de software que pueden usar para sus estudios, proyectos y ocio, promoviendo la cultura colaborativa y el empoderamiento digital.

Objetivos de Aprendizaje

- Comparar las características principales del software libre y el software Open Source.
- Analizar ejemplos reales de programas basados en software libre y Open Source.
- Argumentar la importancia del acceso abierto al código en el desarrollo tecnológico y social.
- Crear un esquema visual que represente las diferencias y similitudes entre ambos conceptos.

Recursos Necesarios

- Computadoras con acceso a internet (1 por estudiante o pareja).
- Proyector y pantalla para presentación multimedia.
- Video breve sobre software libre y Open Source (3-4 minutos).
- Presentación digital con definiciones y ejemplos clave.
- Hojas tamaño carta y marcadores para elaboración de esquemas.
- Cuaderno o dispositivo para tomar notas.
- Encuesta digital (Google Forms o similar) para activación inicial.

Requisitos Previos

- Conocimiento básico sobre qué es un programa o aplicación informática.
- Habilidad para buscar información en internet de manera guiada.
- Experiencia previa en trabajo colaborativo en parejas o pequeños grupos.
- Uso básico de herramientas digitales como navegador web y procesador de texto.

Actividades

Fase de Inicio

Tiempo estimado: 10 minutos

Propósito de la sesión:

Docente: Explica a los estudiantes que hoy explorarán dos conceptos clave para entender cómo funciona gran parte del software que usan: software libre y Open Source. Destaca que conocer estas ideas les ayudará a ser usuarios más críticos y responsables de la tecnología.

Estudiantes: Escuchan y se preparan para participar activamente.

Activación de conocimientos previos:

Docente: Lanza la pregunta para toda la clase: “¿Saben qué es un software? ¿Han oído hablar alguna vez de software que cualquiera pueda usar o modificar?” Luego, invita a responder una encuesta rápida digital con dos preguntas:

- ¿Usas aplicaciones como Firefox, VLC o Android? (Sí/No)
- ¿Crees que el código de esas aplicaciones puede ser visto o modificado por cualquiera? (Sí/No/No sé)

Estudiantes: Responden la encuesta en sus dispositivos. El docente muestra un resumen rápido de las respuestas para conectar con el tema.

Motivación y enganche:

Docente: Comparte un dato curioso: “¿Sabían que el sistema operativo que probablemente usan en sus teléfonos, Android, es un ejemplo de Open Source? Esto significa que su código está abierto para que miles de programadores en el mundo lo mejoren y adapten.”

Pregunta: “¿Qué ventajas creen que tiene que un programa sea abierto para todos?”

Estudiantes: Responden y reflexionan brevemente, generando interés para profundizar.

Contextualización:

Docente: Relaciona el tema con su vida cotidiana: “Muchas de las herramientas que usan para estudiar, comunicarse o divertirse están basadas en software libre o Open Source. Saber cómo funcionan y por qué existen les da poder para usarlas mejor, elegir opciones y hasta crear sus propias soluciones en el futuro.”

Estudiantes: Comprenden la relevancia práctica del tema y se preparan para el desarrollo.

Fase de Desarrollo

Tiempo estimado: 40 minutos

Presentación del contenido:

Docente: Presenta una diapositiva clara y sencilla con definiciones y características clave de software libre y Open Source, usando lenguaje accesible y apoyándose en imágenes y esquemas. Explica:

- Software libre: la libertad de usar, estudiar, modificar y compartir el software.
- Open Source: enfoque en la transparencia del código para mejorar calidad y colaboración.
- Similitudes y diferencias básicas.

Invita a los estudiantes a tomar notas y hacer preguntas.

Actividad 1: Comparación en parejas

- **Objetivo:** Comparar las características principales del software libre y Open Source.

- **Instrucciones:**

Docente: Divide a los estudiantes en parejas y entrega una tabla con dos columnas: “Software Libre” y “Open Source”. Pide que, usando la presentación y sus notas, escriban en cada columna al menos 3 características o ejemplos que recuerden.

Estudiantes: Trabajan en parejas para completar la tabla.

- **Organización:** Parejas
- **Producto:** Tabla comparativa escrita.
- **Tiempo estimado:** 15 minutos
- **Rol del docente:** Circula entre parejas para responder dudas, hacer preguntas como “¿Por qué piensan que esta característica es importante?”, y animar a la reflexión.

Actividad 2: Análisis de casos reales en grupos

- **Objetivo:** Analizar ejemplos reales de programas basados en software libre y Open Source.

- **Instrucciones:**

Docente: Forma grupos de 3-4 estudiantes. Asigna a cada grupo uno de los siguientes programas o proyectos: Linux, Firefox, VLC, WordPress o Wikipedia. Proporciona enlaces o breves descripciones para investigar.

Pide que respondan: ¿Qué es? ¿Por qué es un ejemplo de software libre o Open Source? ¿Cómo puede afectar positivamente a usuarios y desarrolladores?

Estudiantes: Buscan información en línea y discuten para preparar una breve explicación.

- **Organización:** Grupos de 3-4
- **Producto:** Breve presentación oral o por escrito (3-4 frases) explicando su caso.
- **Tiempo estimado:** 20 minutos

- **Rol del docente:** Orienta la búsqueda, formula preguntas guía (“¿Quién puede beneficiarse de este software?”), y motiva la participación de todos.

Actividad 3: Creación de esquema visual

- **Objetivo:** Crear un esquema visual que represente las diferencias y similitudes entre software libre y Open Source.
- **Instrucciones:**
 - Docente:** Pide que en sus grupos o parejas usen hojas y marcadores para diseñar un esquema (diagrama de Venn, mapa conceptual, tabla visual) que muestre lo que aprendieron.
 - Estudiantes:** Elaboran el esquema, usando colores y dibujos para facilitar la comprensión.
- **Organización:** Grupos o parejas
- **Producto:** Esquema visual entregado al docente o expuesto en aula.
- **Tiempo estimado:** 5 minutos
- **Rol del docente:** Observa, ofrece retroalimentación inmediata y destaca ideas clave.

Diferenciación:

- Para estudiantes que terminan antes: Proponerles que investiguen un software libre o Open Source adicional y preparen una breve recomendación para compartir.
- Para estudiantes que necesitan más apoyo: Facilitar resúmenes escritos simples, ofrecer apoyo individual, permitir uso de ejemplos concretos y lenguaje más sencillo.

Transiciones:

Docente: Conecta cada actividad explicando cómo la anterior les ayuda a entender mejor la siguiente y refuerza la importancia de construir conocimiento paso a paso.

Fase de Cierre

Tiempo estimado: 10 minutos

Síntesis:

Docente: Solicita a cada estudiante que escriba en una ficha rápida tres ideas clave que aprendieron sobre software libre y Open Source. Luego, recoge algunas para leerlas en voz alta y construir un mapa mental colectivo en la pizarra o pantalla digital.

Estudiantes: Reflexionan y escriben sus ideas, participan en la construcción colectiva.

Reflexión metacognitiva:

Docente: Formula estas preguntas para que los estudiantes respondan por escrito o en voz alta:

- ¿Cuál es la diferencia más importante entre software libre y Open Source que aprendí hoy?
- ¿Cómo puedo aplicar este conocimiento en mi uso diario de tecnología?
- ¿Qué nueva pregunta tengo sobre el software que quiero investigar más adelante?

Estudiantes: Responden y comparten sus reflexiones.

Retroalimentación:

Docente: Proporciona retroalimentación oral inmediata destacando logros y aclarando dudas comunes basándose en las respuestas y productos de las actividades.

Transferencia:

Docente: Explica que en futuras sesiones explorarán más herramientas tecnológicas y proyectos colaborativos, y anima a los estudiantes a observar qué software usan en casa o en línea que sea libre u Open Source.

Tarea o reto:

Docente: Propone que investiguen en casa un programa o aplicación que sea software libre o Open Source y preparen una breve reseña para compartir en la próxima clase.

Estudiantes: Aceptan el reto para continuar el aprendizaje fuera del aula.

Evaluación

Tipo de evaluación: Diagnóstica al inicio mediante encuesta rápida; formativa durante desarrollo con observación y productos de actividades; sumativa en cierre con síntesis y reflexión.

Criterios de evaluación:

- Identifica correctamente características del software libre y Open Source (objetivo 1).
- Analiza y explica ejemplos reales de software libre y Open Source (objetivo 2).
- Argumenta la importancia del acceso abierto en sus respuestas y discusiones (objetivo 3).
- Elabora un esquema visual claro y representativo que muestre diferencias y similitudes (objetivo 4).

Instrumentos sugeridos:

- Lista de cotejo para evaluar tablas comparativas y esquemas.
- Rúbrica para presentaciones orales o escritas de casos reales.
- Observación directa durante las actividades y discusiones.
- Portafolio digital o físico con productos generados.
- Autoevaluación mediante reflexión escrita al cierre.

Evidencias de aprendizaje:

- Tabla comparativa completada en parejas.
- Presentación o resumen de casos reales en grupos.
- Esquema visual elaborado.
- Respuestas escritas en la reflexión metacognitiva y síntesis.