

Rúbrica para evaluar proyectos de Scratch

2013-08-01

Rúbrica para evaluar proyectos de Scratch

Las rúbricas o matrices de valoración son herramientas valiosas para los docentes pues les ayudan tanto a guiar el trabajo de los estudiantes (proceso) como a valorar la calidad de los productos que han ejecutado (resultado). Ofrecemos un modelo de rúbrica que esperamos les sirva no solo para establecer los criterios que los estudiantes deben cumplir cuando trabajan con Scratch, sino como pauta que les permita a estos últimos comparar sus propios avances en la realización de proyectos de programación.

Las rúbricas o matrices de valoración son herramientas valiosas para los docentes pues les ayudan tanto a guiar el trabajo de los estudiantes (proceso) como a valorar la calidad de los productos que han ejecutado (resultado). Ofrecemos un modelo de rúbrica que esperamos les sirva no solo para establecer los criterios que los estudiantes deben cumplir cuando trabajan con Scratch, sino como pauta que les permita a estos últimos comparar sus propios avances en la realización de proyectos de programación.

•

RÚBRICA PARA EVALUAR PROYECTOS DE SCRATCH

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 solo 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 solo 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%

|

__ Diligencia correctamente los 5 elementos que componen la plantilla de análisis de problemas:

1. Formular el problema;

2. Resultados esperados;

3) Datos disponibles;

4. Restricciones;

5. procesos necesarios [1]

__ Utiliza productivamente el tiempo asignado para realizar el proyecto. Lo culmina antes del plazo de entrega estipulado.

__ Colabora con sus compañeros, incluso, fuera del tiempo de clase.

|

__ Diligencia correctamente 4 de los 5 elementos que componen la plantilla de análisis de problemas:

1. Formular el problema;
- 2) Resultados esperados; 3) Datos disponibles;
4. Restricciones;
5. procesos necesarios [1]

__ Utiliza productivamente el tiempo asignado para realizar el proyecto. Cumple con el plazo de entrega de este.

__ Colabora adecuadamente con sus compañeros de clase.

|

__ Diligencia correctamente 3 de los 5 elementos que comprende la plantilla de análisis de problemas:

1. Formular el problema;
2. Resultados esperados; 3) Datos disponibles;
- 4) Restricciones;
- 5) procesos necesarios [1]

__ La mayoría del tiempo de clase lo utiliza para realizar el proyecto. Cumple con dificultad el plazo de entrega.

__ Colabora con sus compañeros de clase en pocas ocasiones.

|

__ Diligencia correctamente menos de 3 de los 5 elementos que comprende la plantilla de análisis de problemas:

1. Formular el problema;

2. Resultados esperados; 3) Datos disponibles;

3. Restricciones;

5) procesos necesarios [1]

__ No utiliza productivamente el tiempo asignado para realizar el proyecto. No cumple con el plazo de entrega.

__ No colabora con sus compañeros de clase.

|

|

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%

|

__ El programa realizado está organizado, tiene varios niveles y su diseño es complejo.

__ La interfaz gráfica es clara, tiene estructura y se adapta tanto al contenido como al diseño del programa.

__ Es fácil interactuar con el programa.

|

__ El programa realizado está organizado, tiene dos niveles y su diseño es medianamente complejo.

__ La interfaz gráfica es clara pero tiene poca relación con el contenido y con el diseño del programa.

__ Es fácil interactuar con el programa.

|

__ El programa realizado está poco organizado, tiene un solo nivel y su diseño es simple/sencillo.

__ La interfaz gráfica es poco clara y tiene escasa relación tanto con el contenido como con el diseño del programa.

__ Es difícil interactuar con el programa.

|

__ El programa realizado no está organizado y su diseño es básico.

__ La interfaz gráfica es confusa.

__ 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.

|

|

Programación

|

20%

|

__ El programa evidencia comprensión avanzada de bloques y procedimientos.

__ Utiliza apropiadamente las estructuras de control (secuencial, condicional, iterativa).

__ Los hilos de programación son lógicos y están bien organizados.

__ El programa está correctamente depurado.

|

__ El programa demuestra comprensión de los bloques y de cómo estos funcionan en conjunto para alcanzar el resultado esperado.

__ Utiliza apropiadamente algunas estructuras de control (secuencial, condicional, iterativa).

__ Los hilos de programación son lógicos y están organizados.

__ El programa está depurado.

|

__ El programa demuestra alguna comprensión de los bloques y cómo éstos funcionan en conjunto.

__ Utiliza deficientemente las estructuras de control (secuencial, condicional, iterativa).

__ Los hilos de programación tienen poca organización.

☐ El programa tiene una falla de lógica.

|

☐ El programa demuestra poca comprensión de los bloques y de cómo éstos funcionan en conjunto.

☐ Utiliza equivocadamente las estructuras de control (secuencial, condicional, iterativa).

☐ Los hilos de programación carecen de organización.

☐ El programa tiene varias fallas de lógica.

|

|

Pensamiento Computacional [2]

|

15%

|

La elaboración del programa evidencia más de 2 características del pensamiento computacional:

☐ Recopila datos

☐ Analiza datos

☐ Representa datos

☐ Hace abstracciones

☐ Automatiza procesos

☐ Simula procesos

☐ Ejecuta tareas en paralelo

|

La elaboración del programa evidencia 2 características del pensamiento computacional:

__ Recopila datos

__ Analiza datos

__ Representa datos

__ Hace abstracciones

__ Automatiza procesos

__ Simula procesos

__ Ejecuta tareas en paralelo

|

La elaboración del programa evidencia 1 características del pensamiento computacional:

__ Recopila datos

__ Analiza datos

__ Representa datos

__ Hace abstracciones

__ Automatiza procesos

__ Simula procesos

__ Ejecuta tareas en paralelo

|

La elaboración del programa no evidencia características del pensamiento computacional.

|

|

Publicación

|

5%

|

__ El programa se encuentra publicado en la cuenta que el estudiante tiene en el sitio Web de Scratch.

__ En la opción "Notas del Proyecto", están completos los datos que identifican el programa en Scratch:: nombre del estudiante que lo elaboró, nombre de la Institución Educativa, grado escolar del estudiante, asignatura/materia a la que corresponde el proyecto, y corta descripción del programa.

|

|

El programa se encuentra publicado en una cuenta cualquiera del sitio Web de Scratch.

|

__ El programa NO está publicado en el sitio Web de Scratch.

__ En la opción "Notas del Proyecto", NO están completos los datos que identifican el programa en Scratch.

|

|

Contenido del área correspondiente

|

20%

|

Hace conexiones entre los conceptos del tema correspondiente al área para la que se realiza el proyecto. Demuestra comprensión profunda.

|

Involucra en el programa conceptos importantes sobre el tema correspondiente al área para la que se realiza el proyecto.

|

Los conceptos incluidos en el programa tienen poca relación con el tema correspondiente al área para la que se realiza el proyecto.

|

No incluye conceptos sobre el tema del área para la que se realiza el proyecto o, los conceptos son incorrectos.

|

Además de la versión [PDF](#) de esta Matriz de Valoración, también ofrecemos una versión en formato de [Hoja de Cálculo](#) que los docentes pueden utilizar para calificar los trabajos de sus estudiantes.

NOTAS DEL EDITOR:

[1] Con el fin de facilitar el proceso de análisis de problemas, en la Fundación Gabriel Piedrahita Uribe hemos venido promoviendo el uso de una Plantilla de Análisis diseñada especialmente para este fin. Dicha Plantilla tiene dos versiones, una simplificada que se usa con los estudiantes de grados 1° a 3° de básica primaria y otra completa, que se usa con estudiantes de grado 4° en adelante. Es muy importante pedir a los estudiantes que diligencien la Plantilla, de manera sistemática, cada vez que se enfrenten a la solución de un problema. Recomendamos consultar el artículo [Herramienta para analizar problemas](#).

[2] Los avances en computación han ampliado nuestra capacidad para resolver problemas a una escala nunca antes imaginada; por lo tanto, es urgente que los estudiantes no solo aprendan sino que practiquen nuevas habilidades como las de Pensamiento Computacional. Para atender su desarrollo, ISTE y la CSTA aunaron esfuerzos y elaboraron la Caja de Herramientas para Pensamiento Computacional que sugiere cómo trabajarlo en los diferentes grados de la educación escolar. Recomendamos consultar el documento [Pensamiento Computacional, caja de herramientas](#).

CRÉDITOS:

Documento elaborado por EDUTEKA con base en la experticia desarrollada por la Fundación Gabriel Piedrahita Uribe en las cualificaciones a docentes sobre uso pedagógico de Scratch. También se consultaron los siguientes documentos:

[Rubric for assessing Scratch projects](#); Karen Randall & Natalie Rusk, 2009.

- [Creating Scratch Rubrics](#); Rosemary Slattery, 2012.
- [Scratch Project Rubrics](#)

*Publicación de este documento en EDUTEKA: Agosto 01 de 2013.

Última modificación de este documento: Agosto 01 de 2013.*