

Graficador Interactivo de Expresiones Matemáticas

1. Introducción

Las expresiones (léase “fórmulas”) matemáticas pueden ser escritas de muchas formas, incluyendo la infija (la forma en que las escribimos normalmente), prefija (los símbolos van delante, seguidos de los operandos), o sufija (los operadores van al final, precedidos de los operandos). Independiente de ello, los computadores y calculadoras interpretan estas notaciones y las transforman internamente en su equivalente numérico, es decir, resuelven la fórmula y obtienen un resultado.

Para el proyecto, la idea es que el usuario de la aplicación pueda ingresar los números y los operadores en el orden deseado, y que la aplicación interprete los nuevos ingresos y los vaya graficando en tiempo real.

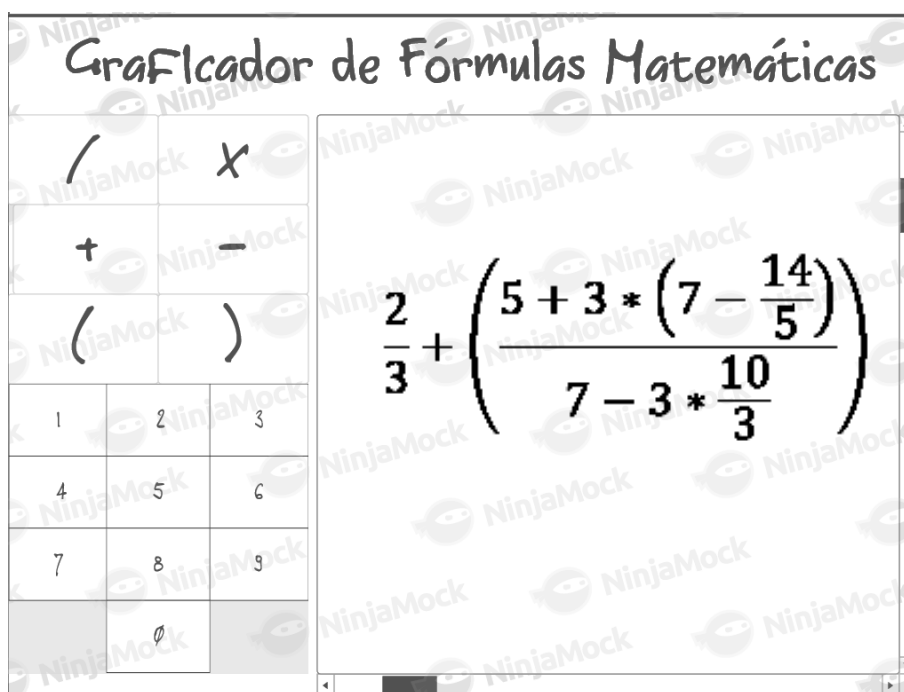


Imagen 1: Ejemplo de interfaz de la aplicación.

La aplicación que deben crear no necesariamente debe tener un interfaz igual al mostrado en la *Imagen 1*; ustedes tienen completo control sobre su diseño de interfaz, siempre y cuando cumplan con el objetivo y los requisitos. Todo depende de cómo quieran abordar el problema.

La precedencia de los operadores matemáticos determina la prioridad de agrupación de los números al dibujar la expresión; por ejemplo, si a la fórmula “3 + 4 * 2” le

agregamos “/4”, la fórmula resultante podría ser $3 + \frac{4*2}{4}$ ó $3 + 4 * \frac{2}{4}$, dependiendo de cómo ustedes definan el agrupamiento de las fracciones. También pueden usar los paréntesis para diferenciar cuáles elementos van en el numerador y cuáles en el denominador de la fracción en caso la expresión.

2. Objetivo

Diseñar e implementar una aplicación que permita “dibujar” de manera interactiva una expresión matemática con operaciones básicas, de tal manera que se muestre gráficamente en un panel de manera similar a la expresión mostrada en la *Imagen 1*.

Cada elemento debe ser “dibujado” en el panel usando primitivas gráficas, en respuesta a eventos del interfaz. Pueden usar cualquier lenguaje de programación y/o librería gráfica, con la salvedad de que sólo podrán usar las clases de dibujo primitivas de línea y/o curva. Sin embargo, podrán extender estas clases para acomodarlas a sus necesidades.

3. Requisitos Indispensables

Los requisitos indispensables son intransables, lo que significa que cumplir con ellos es necesario para aprobar la unidad. No cumplir con al menos uno de ellos es causal de reprobación de la unidad correspondiente, y potencialmente del curso (ver sección de *Observaciones*).

Los requisitos indispensables cumplidos aportan el puntaje indicado, para un total de 3.0 puntos. La nota final de la unidad se calcula contabilizando el puntaje total alcanzado con los requisitos indispensables. En caso de cumplirse todos los requisitos indispensables, puede añadirse el puntaje de los requisitos deseables, usando el esquema descrito en la siguiente sección.

1. La aplicación debe cumplir con los requisitos indispensables y deseables de las unidades anteriores. No cumplir con uno o más de ellos aplicará una penalidad del 20% sobre su nota final de la unidad. Los cambios o aclaraciones a estos requisitos **se incluyen en negrita**.
 - a. Usar repositorio de código (Git).
 - b. Mantener actualizado un tablero KanBan (ej., Trello).
 - c. Definir y mantener un diseño que guíe el desarrollo del proyecto, y que se corresponda con lo que implementan.
 - d. Ingresar interactivamente expresiones matemáticas y dibujarlas en un control, de acuerdo con lo especificado en la sección *Objetivo*. **Debe poder ingresar números de hasta 10 cifras, incluyendo números negativos. Debe validar el no ingreso de símbolos de operaciones seguidos, exceptuando aquellos que si son válidos en matemática (paréntesis, signo menos de los números negativos, etc.).**
 - e. Dibujar los números del 0 al 9 y los operadores aritméticos básicos (+ - / *).
 - f. Mostrar puntos de control **dinámicamente, usando un botón que pueda activarse y desactivarse.**

- g. Asignar colores **elegidos por el usuario** a los números y a los operadores.
2. Implementar el botón “limpiar” (CE), que regresa el programa a su estado inicial (borra el contenido del control de visualización). **(0.50 pto.)**
 3. Agregar operadores de potencia, factorial, seno, coseno, y tangente, mostrando su equivalente dibujado dentro de la fórmula. Distinga números de grados. **(0.75 pto.)**
 4. Mostrar en un panel independiente y visible (ej., un TextBox) la secuencia de números y operadores ingresados, respetando la precedencia de los operadores. **(0.50 pto.)**
 5. Agregar los símbolos y lógica de los paréntesis. Cuando se usen en divisiones, los paréntesis deben “crecer” para cubrir la división y los otros números de la expresión. **(1.25 pto.)**

4. Requisitos Deseables

Los requisitos deseables son opcionales, y por lo tanto su cumplimiento no está exigido. Sin embargo, aportan nota adicional sobre la obtenida por cumplir los requisitos indispensables, siempre y cuando éstos hayan sido cumplidos en su totalidad.

Cada requisito opcional añade puntaje a la nota final del proyecto, con un máximo de 3.0 puntos, para una nota potencial total de 7.0. Nótese que, de no cumplir con los requisitos indispensables de la Unidad, el puntaje obtenido con los requisitos deseables se perderá; es decir, no contará para la nota final de proyecto.

1. Agregar una opción que permita cambiar el modo de la calculadora entre Básico y Científico. El modo Básico debe mostrar solamente los operadores de la Unidad I, y el modo Científico debe además incluir los operadores de la Unidad II. Esto significa que deben variar la distribución de botones al cambiar de modo, y de ser necesario la apariencia del panel principal (ej., hacer más grande el panel de botones, y/o el área de graficado). **(0.75 pto.)**
2. Agregar botones que permitan cambiar de base los números (Base 10, Base 2). Los números del panel de visualización (el canvas) deben cambiar automáticamente al modificar la base, pero no los del panel de expresión matemática (el textbox). **(1.0 pto.)**
3. Incluir opciones de cambio de tamaño de las expresiones ingresadas. Deben poder reducir o agrandar los elementos dibujados sin afectar la visualización de estos (osea, no solo agrandar o reducir los símbolos, sino también el espacio entre ellos). Lo pueden hacer a través de un menú (“muy pequeño, pequeño, mediano, grande, muy grande”, “7 8, 9, 10, 11”, etc.) que tenga al menos 5 opciones de tamaños notoriamente distintos, o con botones (+, -) que incrementen y disminuyan el tamaño dentro de un rango (máximo y mínimo). No cuenta hacer zoom al canvas; deben transformar o redibujar los elementos. **(1.25 pto.)**

5. Potenciales requisitos futuros

Los requisitos deseables de una unidad siempre son exigibles en la unidad siguiente, por lo que sirve hacer las cosas hoy para tener tiempo libre mañana. Aparte de ello, los siguientes requisitos podrían ser considerados para unidades futuras tal cual aparecen listados, modificados, o incluso podrían ser eliminados. Vale la pena tomarlos en cuenta para su diseño, pero no se le ocurra implementarlos... o podría terminar trabajando en vano.

- Calcular el resultado de una operación ingresada por el usuario.
- Agregar un operador de raíz cuadrada que dibuje su representación gráfica $(\sqrt{2 + \frac{3}{4*5}})$.
- Cambiar el orden de la secuencia de símbolos ingresados en el panel entre los tres esquemas de ordenamiento de árboles, postorden, enorden, y preorden. Por ejemplo, si ingresan la expresión $4*3/(4+5)$, debería mostrarse $4*(3/(4+5))$ en inorden, $(/ (*4 3) (+ 4 5))$ en preorden, y $((4 3 *) (4 5 +) /)$ en postorden. Revisar lógica de recorrido de árboles para más información.
- Agregar los botones maximizar y minimizar, y sus funcionalidades correspondientes.
- Panel de dibujo con tamaño variable.

6. Fechas Importantes

Fecha de Inicio	:	29/04/2022
Fecha de Término	:	30/05/2022
Extensión	:	06/06/2022
Revisiones	:	En horario de clase.

7. Observaciones

- La nota de proyecto sólo representa el 50% de la nota final de la unidad. El otro 50% corresponde a una rúbrica de desempeño individual, que evalúa la participación, aporte, conocimiento de la aplicación, y nivel y calidad de código presentado por cada integrante.
- Durante las presentaciones de unidad, sólo tendrán derecho a *dos caídas de aplicación* (excepción, corrupción de datos, cerrado manual, etc.). Si su programa se cierra una tercera vez, reprobarán la unidad o recuperación correspondiente. ¡Valide sus entradas y salidas!
- El enunciado del proyecto (v1.0) es preliminar, y está sujeto a cambios menores. En el transcurso de las siguientes semanas podría ser especificado formalmente, incluyendo la rúbrica, y posibles modificaciones a los requisitos indispensables y opcionales.
- El proyecto de la unidad III es recuperable, en la forma de tiempo adicional (hasta una semana). Aprobar una recuperación permite optar como máximo a un 4.0 en la nota de proyecto. En caso de reprobación, quedarán con su nota correspondiente. La nota de rúbrica no es recuperable... ¡mucho cuidado!
- La unidad III es exigible, por lo que reprobación del proyecto de dicha unidad (y su recuperación) implica reprobación del curso. Recuerde que también puede reprobación por puntaje, así que apunte alto. ¡No se confíe!

Última actualización: 29/04/2022