

# Lineamientos técnicos

Algunos equipos nos han señalado la existencia de bibliotecas y frameworks que permiten el modelado y evaluación de expresiones matemáticas. Si bien estas herramientas funcionan y podrían ser útiles, las operaciones **no deberán ser modeladas en torno a éstas**; en su lugar los equipos deberán modelar las operaciones como elementos del dominio, independientes de cualquier componente externo.

El motivo es que no queremos delegar el comportamiento **principal** de nuestro dominio a un componente desarrollado por un tercero. De lo contrario:

1. No tendríamos control sobre los cambios de dicho componente:
  - a. Sus cambios podrían afectar severamente a nuestro sistema.
  - b. Si encontráramos un *bug* en éste, no podríamos corregirlo fácilmente
2. Se limitaría fuertemente la flexibilidad de nuestro sistema, dado que no podríamos quitar, modificar o agregar comportamiento nuevo a estos componentes.
3. La mayoría de estas herramientas basan su funcionamiento en una representación textual particular de las expresiones. Sin embargo, en nuestro sistema esa representación textual es accidental, y podría cambiar o ser reemplazada por [otras formas de representación](#).

Además, didácticamente, nos permite analizar problemas de diseño más ricos.

## Sobre la complejidad de las expresiones

Como ya sabemos, los indicadores son expresiones matemáticas que están en función de cuentas y otros indicadores. Si bien idealmente queremos poder soportar expresiones matemáticas arbitrarias:

- Por un lado esto es muy complejo y requiere gran cantidad de tiempo de desarrollo, además de que el alcance no está bien definido (¿tendremos que soportar multiplicaciones, logaritmos, derivaciones, integrales?);
- Por otro lado, las expresiones matemáticas utilizadas en este dominio son normalmente muy limitadas: multiplicaciones, cocientes, sumas y restas; en definitiva [operaciones algebraicas](#).

Por tanto, nuestras prioridades serán soportar fórmulas del estilo:

1.  $\text{IndicadorXX} = 1$
2.  $\text{IndicadorXX} = \text{CUENTA}$
3.  $\text{IndicadorXX} = \text{CUENTA} \oplus \text{VALOR}$
4.  $\text{IndicadorYX} = \text{IndicadorXX} \oplus \text{VALOR} \oplus \text{CUENTA}$

Siendo  $\oplus$  una operación algebraica, entre un valor, una cuenta u otro indicador.

# Sobre la UI

La interfaz de usuario para el ingreso de indicadores debe poder ingresarlos de alguna de las dos siguientes forma, a criterio del equipo:

- En modo texto, es decir, mediante un string que represente textualmente la operación como "X + 2". Esta es la opción **recomendada**
- Mediante una interfaz gráfica, que permita crear las operaciones mediante botones, combos y cualquier otro componente visual que lo requiera. Esta forma alternativa **también es aceptable**.

**En cualquier caso**, se debe:

- Tener en cuenta y documentar ventajas y desventajas de cada opción
- Contemplar que el usuario puede equivocarse e ingresar fórmulas incorrectas (mal formadas). En tal caso, el sistema debe reportar adecuadamente la situación.

Ejemplos:

- `IndicadorA = CUENTA * 3 +`
- `IndicadorB = IndicadorB`

# Sobre la persistencia

Esto es al cliente le será útil, pero es prioritario todo lo anterior y sin ello no podrá darnos un feedback del funcionamiento del sistema.

Como durante esta parte del cuatrimestre no profundizamos sobre los mecanismos de persistencia, se les recuerda que pueden ir por algo tan simple como un archivo en el formato deseado.