

Refactoring Ejemplo 1

Por: Juan Sebastián Sánchez L.

Refactoring: Modificación de funciones

- Descripción: Se crearon funciones separadas *amountFor*, *calculateVolumeCredits*, y *format* para manejar cálculos y formateo específicos.
- Razón de aplicación: Separar funcionalidad en funciones independientes hace que el código sea más modular y más fácil de mantener.
- Consecuencia: El código se vuelve más organizado y legible, y cada función tiene una responsabilidad clara.

Refactoring: Organización de información

- Descripción: Se organizó la lógica relacionada con cálculos y formateo en funciones separadas.
- Razón de aplicación: Organizar el código de esta manera facilita la gestión y el mantenimiento de las distintas partes del programa.
- Consecuencia: El código es más modular y estructurado, lo que mejora la mantenibilidad.

Refactoring: Modificación de funciones

- Descripción: Se utilizó la función *amountFor* para realizar los cálculos específicos de cada tipo de obra de teatro.
- Razón de aplicación: Dividir la lógica de cálculos específicos en funciones separadas mejora la modularidad y reduce la duplicación de código.
- Consecuencia: El código se vuelve más claro y mantenible al separar los cálculos en funciones específicas.

Refactoring: Organización de información

- Descripción: Se organizó la generación del resultado en una función independiente llamada *generateStatement*.
- Razón de aplicación: Separar la generación del resultado del cálculo de lógica de negocio mejora la estructura del programa.
- Consecuencia: El código es más claro y enfocado en la lógica de negocio, mientras que la generación del resultado se maneja en una función independiente.