

Relatório de Manutenção do SHA V2

Aluno proprietário: Arthur Siqueira

Aluno colaborador: Cefras Mandú

Prof.: Katysco Santos

Este relatório detalha a arquitetura e a operação de um simulador de hidrômetro desenvolvido por Arthur em Java. O sistema é composto por três classes principais: Simulador, Hidrômetro e Display, que trabalham em conjunto para simular o consumo de água e exibir os dados de leitura de forma visual.

Onde inicialmente para realizar a manutenção, foi identificado que a versão do simulador possuía a contagem do consumo total de água e era feita a atualização da imagem em um único arquivo no formato de imagem, porém grande parte dos requisitos do simulador não estavam sendo atendidos, como os parâmetros para que seja feita a simulação de forma ininterrupta e que fosse possível aumentar ou diminuir a vazão de água sem que fosse necessário parar a simulação, a amostragem de forma visual do funcionamento do hidrômetro, além do registro da contagem a cada metro cúbico sendo armazenado na pasta de medição.

Devido a esses pontos foi necessário ser corrigido e implementado essas mudanças para que o projeto possa ser executado corretamente, e adicionado também o novo requisito de simulação simultânea. As mudanças na estrutura do projeto estarão descritas posteriormente nesse relatório para que fim de comparação entre a diferença da versão que foi concedida o acesso ao projeto para essa versão final.

1. Arquitetura e Componentes

O sistema é modelado conforme o diagrama de classes UML fornecido, que estabelece as seguintes relações:

- **Simulador:** É a classe principal que orquestra a simulação. Ela cria instâncias de Hidrômetro e Display e gerencia o fluxo de execução.
- **Hidrômetro:** Representa o medidor de água. É responsável por armazenar o número de série e contabilizar o volume de água consumido.
- **Display:** Responsável pela interface gráfica do simulador. Gera uma imagem que exibe os dados atuais do hidrômetro.

As relações entre as classes são:

- **Associação:** A classe Simulador tem uma associação com Hidrômetro e Display, indicando que ela cria e mantém instâncias desses objetos para operar.
- **Dependência:** A classe Display depende da classe Hidrômetro, pois precisa de um objeto Hidrômetro para obter os dados necessários para gerar a imagem.

2. Descrição Detalhada das Classes

2.1. Classe Hidrometro

Esta classe encapsula os dados e as operações de um hidrômetro real.

- **Atributos:**
 - numeroSerie (String): Armazena o número de série do hidrômetro.
 - litrosConsumidos (int): Acumula o total de litros de água consumidos.
- **Métodos:**
 - Hidrometro(String numeroSerie): Construtor que inicializa o objeto com um número de série e zera a contagem de litros consumidos.
 - consumirAgua(int litros): Adiciona um valor ao total de litros consumidos.
 - getLitrosConsumidos(): Retorna o total de litros consumidos.
 - getNumeroSerie(): Retorna o número de série do hidrômetro.
 - getMetrosCubicos(): Calcula e retorna a parte inteira do consumo em metros cúbicos (litros / 1000).
 - getLitrosRestantes(): Retorna os litros que ainda não completaram 1 metro cúbico (litros % 1000).

2.2. Classe Display

Responsável por gerar uma representação visual dos dados do hidrômetro.

- **Métodos:**
 - gerarImagem(Hidrometro h): Este é o método central da classe. Ele executa as seguintes ações:
 1. Cria uma imagem em branco (BufferedImage) com dimensões de 400x200 pixels.
 2. Define um fundo branco para a imagem.
 3. Desenha o número de série do hidrômetro no topo da imagem.

4. Exibe o consumo total em litros, formatado com seis dígitos.
5. Mostra o consumo convertido para metros cúbicos com duas casas decimais.
6. Exibe os litros restantes e os décimos de litro em seções separadas.
7. Salva a imagem gerada em um arquivo chamado hidrometro_saida.png.
8. Imprime uma mensagem no console informando que a imagem foi atualizada, juntamente com o consumo total.
9. Possui um bloco de tratamento de exceções para capturar possíveis erros durante a manipulação da imagem.

2.3. Classe Simulador

É o ponto de entrada e o controlador do programa.

- **Métodos:**

- `main(String[] args)`: O método principal que executa a simulação.
 1. Instancia um objeto
Hidrometro com o número de série "12345-ABC".
 2. Instancia um objeto Display.
 3. Entra em um loop infinito (`while (true)`) para simular o consumo contínuo de água.
 4. Dentro do loop, a cada ciclo:
 - Gera um número aleatório de consumo entre 1 e 20 litros.
 - Chama o método
`h.consumirAgua()` para atualizar o consumo no hidrômetro.
 - Chama o método `d.gerarImagem()` para atualizar a imagem com os novos dados de consumo.
 - Aguarda por 2 segundos (`Thread.sleep(2000)`) antes de iniciar o próximo ciclo de consumo, controlando a velocidade da simulação.
 5. Inclui um bloco
`try-catch` para tratar a exceção `InterruptedException` que pode ocorrer durante a pausa.

3. Funcionamento da Simulação

1. A simulação é iniciada pela classe Simulador.
2. Um hidrômetro é criado e associado a um display.
3. O programa entra em um ciclo contínuo, onde a cada dois segundos um volume aleatório de água (entre 1 e 20 litros) é consumido.
4. O objeto Hidrometro atualiza seu contador interno de litrosConsumidos.
5. Em seguida, o objeto Display utiliza os dados atualizados do Hidrometro para gerar e salvar um novo arquivo de imagem (hidrometro_saida.png), refletindo o consumo atualizado.
6. Uma mensagem é exibida no console a cada atualização, permitindo o acompanhamento do progresso da simulação em tempo real.
7. Este processo se repete indefinidamente, simulando o funcionamento contínuo de um medidor de água.

4. Evolução do Projeto e Novas Funcionalidades

A versão inicial do projeto serviu como uma base conceitual, implementando a contagem de consumo e a geração de uma imagem de saída. No entanto, para atender aos requisitos especificados, foram necessárias extensas modificações e a adição de novas funcionalidades, transformando o programa de um simples gerador de imagens em um simulador interativo e robusto. A seguir, são detalhadas as principais evoluções, comparando a versão final com a original.

4.1. Mudanças Arquiteturais

A arquitetura do projeto foi completamente redesenhada para suportar os novos requisitos de interatividade e simulação simultânea.

• De Single-Thread para Multi-Thread:

- **Versão Inicial:** O programa rodava em um único thread dentro do método main, executando um laço infinito que simulava e atualizava a imagem.
- **Versão Final:** A arquitetura agora é multi-thread. A classe Simulador atua apenas como um "lançador", que pode instanciar múltiplos objetos SimuladorInstancia. Cada SimuladorInstancia implementa a interface Runnable e encapsula uma simulação completa e independente (lógica, dados e interface gráfica), rodando em seu próprio thread.

• De Configuração Fixa para Externa:

- **Versão Inicial:** Parâmetros como o número de série do hidrômetro eram fixos no código-fonte.

- **Versão Final:** Parâmetros essenciais como vazão inicial, pressão, probabilidades de eventos e a matrícula do aluno para salvamento de medições são carregados de um arquivo externo `parametros.properties`. Isso permite ajustar o comportamento da simulação sem a necessidade de recompilar o código.

- **De Simulação Única para Múltipla:**

- **Versão Inicial:** Apenas uma instância do hidrômetro era simulada.
- **Versão Final:** Foi adicionado o novo requisito de simulação simultânea. Ao iniciar, o programa pergunta ao usuário quantos hidrômetros (de 1 a 5) devem ser simulados, e lança uma janela e uma thread independente para cada um.

4.2. Evolução da Interface Gráfica (Classe Display)

A maior mudança visível foi na classe Display, que evoluiu de um simples gerador de arquivos de imagem para uma interface gráfica interativa (GUI).

- **De Imagem Estática para Janela Interativa:**

- **Versão Inicial:** A classe Display gerava um arquivo `hidrometro_saida.png` a cada 2 segundos, com os dados do consumo escritos como texto simples.
- **Versão Final:** A classe Display agora herda de JFrame e cria uma janela real utilizando a biblioteca Java Swing. Ela utiliza uma imagem de fundo para dar a aparência de um hidrômetro real e é atualizada de forma fluida a aproximadamente 60 quadros por segundo.

- **De Texto Simples para Visualização Analógica:**

- **Versão Inicial:** O consumo era exibido apenas como texto.
- **Versão Final:** A GUI agora possui ponteiros analógicos animados que representam visualmente o fluxo de água (ponteiro vermelho, que gira com o consumo) e a pressão da rede (ponteiro verde). A leitura numérica também foi aprimorada, com formatação 00000,000 e os dígitos fracionários destacados em vermelho.

4.3. Avanços na Lógica de Simulação (Classe Hidrômetro)

A classe Hidrômetro foi aprimorada para suportar uma simulação muito mais detalhada e realista.

- **Novos Parâmetros de Simulação:**

- **Versão Inicial:** A classe apenas contabilizava `litrosConsumidos`⁸.

Hidrômetro Nº 12345-ABC

000068

Consumo: 0,07 m³

Litros restantes: 68

Décimo de litro: 8

- **Versão Final:** A simulação agora inclui **vazão** () e **pressão** (bar) como parâmetros dinâmicos.

- **Sistema de Eventos com Duração:**

- **Versão Inicial:** O consumo era contínuo e sem imprevistos⁹.
- **Versão Final:** Foi implementado um sistema de eventos com duração. O simulador pode entrar em estados de "**SEM FLUXO**" ou "**AR NA TUBULAÇÃO**". Esses eventos permanecem ativos por um período de tempo visível, durante o qual a pressão e a medição são afetadas, e o ponteiro de pressão reage de acordo.

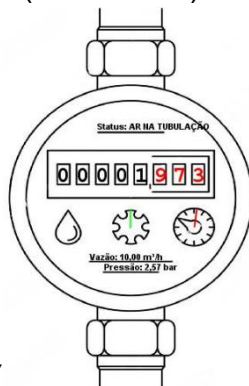
- **Realismo Mecânico:**

- **Versão Inicial:** O fluxo de água era perfeitamente constante a cada ciclo.
- **Versão Final:** Foi adicionada uma leve e aleatória **flutuação** à vazão a cada passo da simulação, imitando as pequenas imperfeições de um sistema mecânico real e tornando o movimento dos ponteiros mais natural e menos uniforme.

4.4. Novas Funcionalidades Interativas e de Requisito

Foram implementadas funcionalidades-chave que não existiam na versão original para atender plenamente aos requisitos.

- **Controle de Vazão Interativo:** O usuário pode aumentar ou diminuir a vazão da água em tempo real através de botões na interface, sem precisar parar ou reiniciar a simulação, atendendo a um dos requisitos principais que estavam ausentes.
- **Salvamento Condicional de Medições:** O sistema agora salva automaticamente uma captura de tela (screenshot) da sua própria interface no formato JPEG. Isso ocorre a cada



metro cúbico (

-) inteiro de água consumido, e as imagens são salvas em um diretório específico (Medições_[Matricula]), cumprindo outro requisito central do projeto.