Roteiro de Atividade Prática

Nome: Turma: .

**Atividade: Design e padrões de mensagens**

**Tempo estimado:** 30 minutos.

**Lista de materiais**

* Computador com internet;
* Caderno para anotações;
* 1 caneta.

**Objetivo**:

Esta atividade visa implementar o padrão *Event Sourcing* para gerenciar o estado de uma conta bancária, onde cada transação é tratada como um evento que pode ser aplicado para alterar o estado da conta.

**Contexto:**

*Event Sourcing* é um padrão de design que captura todas as mudanças em um estado de aplicativo como uma sequência de eventos. Isso não apenas permite recriar o estado do aplicativo a partir desses eventos, mas também oferece um histórico completo e auditável das mudanças ao longo do tempo.

**Enunciado**:

Desenvolva um algoritmo que utilize o padrão *Event Sourcing* para registrar e aplicar eventos (como depósitos e retiradas) a uma conta bancária. Esse algoritmo deve permitir a reconstrução do estado da conta bancária a qualquer momento, baseando-se, exclusivamente, em seu histórico de eventos. Acompanhe as etapas detalhadas abaixo para desenvolver o algoritmo.

1. **Etapas para desenvolver o algoritmo**
   1. **Definição de eventos**:
   * Cada evento deve ter um tipo (por exemplo, ***deposit*** ou ***withdrawal***) e um montante associado;
   * O evento também pode incluir uma data e hora, bem como informações adicionais relevantes, como a identificação do usuário que realizou a transação.
   1. **Estrutura da conta bancária**:

* A conta bancária deve iniciar com um saldo de zero;
* A conta mantém uma lista de todos os eventos que afetaram seu estado.

**1.3 Processo de aplicação de eventos**:

* **Entradas**: Eventos individuais, cada um contendo tipo e montante.
* **Processo**:
* Iterar sobre cada evento no histórico;
* Para cada evento de depósito, adicione o montante ao saldo da conta;
* Para cada evento de retirada, subtraia o montante do saldo, assumindo saldo suficiente para cobrir a retirada.
* **Saída**: O estado atualizado da conta (saldo).  
  1. **Reconstrução do estado**:
* Em qualquer ponto, o estado da conta pode ser reconstruído processando sequencialmente os eventos desde o início;
* Isso permite a verificação do estado da conta em qualquer momento histórico.

1. **Passo a passo da implementação**  
   1. **Definição de classes ou estruturas***:*

* Defina uma classe ou estrutura para os eventos, com campos para tipo, montante e data/hora;
* Defina uma classe para a conta bancária, incluindo uma lista de eventos e um método para aplicar eventos ao saldo.  
  1. **Implementação da lógica de eventos**:
* Implemente métodos para criar e registrar eventos de depósito e retirada;
* Esses eventos devem ser adicionados ao histórico da conta cada vez que uma transação é realizada.  
  1. **Aplicação dos eventos ao estado**:
* Desenvolva um método na classe da conta que processa todos os eventos para atualizar o saldo;
* Esse método pode ser chamado para reconstruir o saldo a qualquer momento, processando eventos desde o início ou desde o último estado salvo.  
  1. **Testes**:
* Crie cenários de teste que incluam depósitos e retiradas variadas;
* Teste a reconstrução do estado em diferentes pontos no tempo para verificar a integridade do *Event Sourcing*.  
  1. **Monitoramento e auditoria**:
* Implemente funcionalidades para facilitar o monitoramento e a auditoria do histórico de transações;
* Isso pode incluir métodos para recuperar o histórico de eventos em diferentes formatos ou estados.

Esse algoritmo permite uma rastreabilidade completa das transações e fornece uma base sólida para sistemas que requerem alta integridade e auditabilidade. Implementar *Event Sourcing* em uma conta bancária demonstra como o estado do sistema pode ser gerido de forma imutável e transparente, oferecendo benefícios significativos em termos de segurança, conformidade e manutenção.

**Procedimento experimental**

1. A partir do contexto das etapas desenvolvidas, descreva o algoritmo obtido na atividade e registre no AVA.

|  |
| --- |
|  |
|  |
|  |
|  |
|  |