PLANO PARA ESPECIFICAÇÃO FORMAL EM NOTAÇÃO Z

PROJETO: Sistema de Gestão Financeira e Simulação de Supermercado

1. ANÁLISE DO DOMÍNIO

1.1 Entidades Principais Identificadas:

- Usuário (User)
- Sessão de Jogo (GameSession)
- Produto (Product)
- Categoria de Produto (ProductCategory)
- Fornecedor (Supplier)
- Transação Financeira (Transaction)
- Saldo do Usuário (UserBalance)
- Histórico de Estoque (ProductStockHistory)
- Venda em Tempo Real (RealtimeSale)

1.2 Operações Críticas:

- Autenticação de usuário
- Gestão de estoque
- Processamento de vendas
- Atualização de saldo
- Controle de tempo do jogo
- Vendas automáticas

2. ESTRUTURA DA ESPECIFICAÇÃO Z

2.1 Esquemas de Estado (State Schemas):

- SistemaGlobal: Estado geral do sistema
- UsuarioSchema: Estado dos usuários
- JogoSchema: Estado das sessões de jogo
- ProdutoSchema: Estado dos produtos
- FinanceiroSchema: Estado financeiro

- EstoqueSchema: Estado do estoque

2.2 Esquemas de Operação (Operation Schemas):

- LoginUsuario
- CriarSessaoJogo
- AdicionarProduto
- RemoverProduto
- ProcessarVenda
- AtualizarSaldo
- AvancarTempoJogo
- ProcessarVendasAutomaticas

2.3 Esquemas de Inicialização:

- InitSistema
- InitUsuario
- InitJogo

3. DETALHAMENTO DOS ESQUEMAS

3.1 Esquemas de Estado:

3.1.1 SistemaGlobal:

- `usuarios`: conjunto de usuários ativos
- `sessoes_jogo`: conjunto de sessões ativas
- `produtos`: conjunto de produtos disponíveis
- `fornecedores`: conjunto de fornecedores
- `categorias`: conjunto de categorias de produtos

3.1.2 UsuarioSchema:

- `id_usuario`: identificador único
- `nome`: nome do usuário
- `email`: email único
- `saldo_atual`: saldo financeiro atual
- `data_criacao`: timestamp de criação

3.1.3 JogoSchema:

- `id_sessao`: identificador da sessão
- `usuario`: referência ao usuário
- `data_inicio_jogo`: data de início no jogo
- `data_atual_jogo`: data atual no jogo
- `data_fim_jogo`: data de fim do jogo
- `status`: estado da sessão (ATIVO, PAUSADO, COMPLETO, FALHOU)

- `aceleracao_tempo`: segundos reais por dia do jogo
- `meta_vendas_diarias`: número de vendas por dia
- `vendas_automaticas_habilitadas`: flag booleana

3.1.4 ProdutoSchema:

- `id_produto`: identificador único
- `nome`: nome do produto
- `categoria`: referência à categoria
- `fornecedor`: referência ao fornecedor
- `preco_compra`: preço de compra
- `preco_venda`: preço de venda
- `estoque_atual`: quantidade em estoque
- `estoque_minimo`: estoque mínimo
- `estoque_maximo`: estoque máximo
- `ativo`: flag de disponibilidade

3.1.5 FinanceiroSchema:

- `id_transacao`: identificador único
- `usuario`: referência ao usuário
- `tipo`: RECEITA ou DESPESA
- `valor`: valor da transação
- `categoria`: categoria da transação
- `data_transacao`: data da transação
- `descricao`: descrição da transação

3.1.6 EstoqueSchema:

- `id_historico`: identificador único
- `produto`: referência ao produto
- `operacao`: tipo de operação (COMPRA, VENDA, AJUSTE, PERDA)
- `quantidade`: quantidade movimentada
- `estoque_anterior`: estoque antes da operação
- `estoque_novo`: estoque após a operação
- `data_operacao`: data da operação

3.2 Esquemas de Operação:

3.2.1 LoginUsuario:

- **Entrada:** email, senha
- **Saída:** token de autenticação, dados do usuário
- **Pré-condições:** email válido, senha correta
- **Pós-condições:** usuário autenticado, sessão criada

3.2.2 CriarSessaoJogo:

- **Entrada:** `id_usuario`
- **Saída:** `id_sessao`
- **Pré-condições:** usuário existe e não possui sessão ativa
- **Pós-condições:** nova sessão criada com configurações padrão

3.2.3 AdicionarProduto:

- **Entrada:** dados do produto
- **Saída:** `id_produto`
- **Pré-condições:** categoria e fornecedor existem
- **Pós-condições:** produto adicionado ao sistema

3.2.4 RemoverProduto:

- **Entrada:** `id_produto`, quantidade
- **Saída:** confirmação
- **Pré-condições:** produto existe e tem estoque suficiente
- **Pós-condições:** estoque reduzido, histórico atualizado

3.2.5 ProcessarVenda:

- **Entrada:** `id_produto`, quantidade, `id_sessao`
- **Saída:** `valor_total`, `nova_data_jogo`
- **Pré-condições:** produto disponível, estoque suficiente, sessão ativa
- **Pós-condições:** estoque reduzido, saldo atualizado, transação criada

3.2.6 AtualizarSaldo:

- **Entrada:** `id_usuario`, valor, `tipo_operacao`
- **Saída:** `novo_saldo`
- **Pré-condições:** usuário existe
- **Pós-condições:** saldo atualizado, histórico criado

3.2.7 AvancarTempoJogo:

- **Entrada:** `id_sessao`, `tempo_real_decorrido`
- **Saída:** `dias_jogo_passados`
- **Pré-condições:** sessão existe e está ativa
- **Pós-condições:** tempo do jogo avançado, vendas processadas

3.2.8 ProcessarVendasAutomaticas:

- **Entrada:** `id_sessao`, `dias_passados`
- **Saída:** `vendas_processadas`
- **Pré-condições:** sessão ativa, vendas automáticas habilitadas
- **Pós-condições:** vendas criadas, estoque e saldo atualizadosINVARIANTES DO SISTEMA

4.1 Invariantes de Integridade:

- Todo usuário deve ter saldo não negativo
- Todo produto deve ter estoque não negativo
- Toda sessão deve ter usuário válido
- Toda transação deve ter usuário válido
- Todo histórico de estoque deve ter produto válido

4.2 Invariantes de Consistência:

- Saldo do usuário = soma de todas as transações
- Estoque atual = estoque inicial + compras - vendas - perdas
- Data atual do jogo >= data de início
- Data atual do jogo <= data de fim

4.3 Invariantes de Negócio:

- Preço de venda >= preço de compra
- Estoque mínimo <= estoque atual <= estoque máximo
- Meta de vendas diárias > 0
- Aceleração de tempo > 0

5. ESPECIFICAÇÕES DE SEGURANÇA

5.1 Controle de Acesso:

- Apenas usuários autenticados podem acessar o sistema
- Usuários só podem modificar seus próprios dados
- Operações administrativas requerem privilégios especiais

5.2 Validação de Dados:

- Todos os valores monetários devem ser positivos
- Quantidades devem ser números inteiros não negativos
- Datas devem estar em formato válido
- Emails devem ter formato válido

6. ESPECIFICAÇÕES DE PERFORMANCE

6.1 Limites Operacionais:

- Máximo de 1000 usuários simultâneos
- Máximo de 10000 produtos no sistema
- Processamento de vendas em tempo real (< 1 segundo)

- Atualização de saldo em tempo real (< 500ms)

6.2 Garantias de Disponibilidade:

- Sistema deve estar disponível 99.9% do tempo
- Backup automático a cada 24 horas
- Recuperação de falhas em menos de 5 minutos

7. PLANO DE IMPLEMENTAÇÃO

7.1 Fase 1 - Esquemas Básicos:

- Definir tipos básicos (ID, Data, Valor, etc.)
- Implementar esquemas de estado principais
- Definir invariantes básicos

7.2 Fase 2 - Operações Core:

- Implementar operações de usuário
- Implementar operações de produto
- Implementar operações financeiras básicas

7.3 Fase 3 - Operações Complexas:

- Implementar sistema de jogo
- Implementar vendas automáticas
- Implementar controle de tempo

7.4 Fase 4 - Validação e Refinamento:

- Validar todas as operações
- Refinar invariantes
- Adicionar especificações de erro

8. FERRAMENTAS E NOTAÇÃO

8.1 Notação Z:

- Usar sintaxe padrão da Notação Z
- Definir tipos básicos claramente
- Usar convenções de nomenclatura consistentes

8.2 Ferramentas Sugeridas:

- Z/EVES para verificação

- LaTeX para formatação
- Fuzz para análise de tipos

9. CRITÉRIOS DE VALIDAÇÃO

9.1 Completude:

- Todas as operações do sistema devem estar especificadas
- Todos os estados possíveis devem ser cobertos
- Todas as transições devem ser definidas

9.2 Consistência:

- Não deve haver contradições entre esquemas
- Invariantes devem ser preservados
- Operações devem ser bem-definidas

9.3 Correção:

- Especificações devem refletir o comportamento real
- Casos de erro devem ser tratados
- Propriedades de segurança devem ser garantidas

10. ENTREGÁVEIS

10.1 Documentos:

- Especificação completa em Notação Z
- Documento de invariantes
- Guia de validação
- Relatório de análise

10.2 Artefatos:

- Arquivos .tex com especificações
- Scripts de verificação
- Casos de teste
- Documentação de API