

Projeto War

Professor: Dr. Diego Pinheiro

Atividade: Projeto da disciplina

Disciplina: Engenharia de Software 2

Projeto

Propor análise e projeto de um software orientado a objetos para o famoso jogo de tabuleiro War. A versão do jogo que deve ser considerada é a versão original que é mais clássica e simples. Todos integrantes devem pesquisar sobre todas as regras do jogo e entender sobre os aspectos relacionados ao domínio do jogo. Jogar o jogo de tabuleiro em equipe ajudará bastante.

Embora os membros da equipe estão livres para colaborar entre si, **cada membro da equipe será avaliado individualmente** em relação ao escopo bem definido que lhe será atribuído pelo qual é inteiramente responsável.

Os escopos serão os seguintes:

1. Inicialização de um novo jogo
2. Todos os objetivos relacionados à conquista de territórios e continentes;
3. Fase de fortificação dos territórios no início de uma jogada
4. Realização de uma jogada e fase de povoamento de um território recentemente conquistado;
5. Fase de ataque entre territórios
6. Fase de realocação dos territórios no fim da jogada e todos os objetivos relacionados à eliminação de jogadores

Sprint 1

Nesta sprint, os artefatos esperados são **diagrama de casos de uso**, **modelo de domínio** e **especificação de casos de uso**, **contrato de operação**, **diagrama de comunicação com projeto de objetos da pré-condição e pós-condição**, e o **diagrama de classes**. Os artefatos esperados devem utilizar como referência as bibliografias indicadas no plano de ensino da disciplina.

Diagrama de Casos de Uso

Confeccionar a primeira versão do **Diagrama de Casos de Uso** identificando os atores e os casos de uso. Consultar as referências bibliográfica no plano de ensino para exemplos concretos.

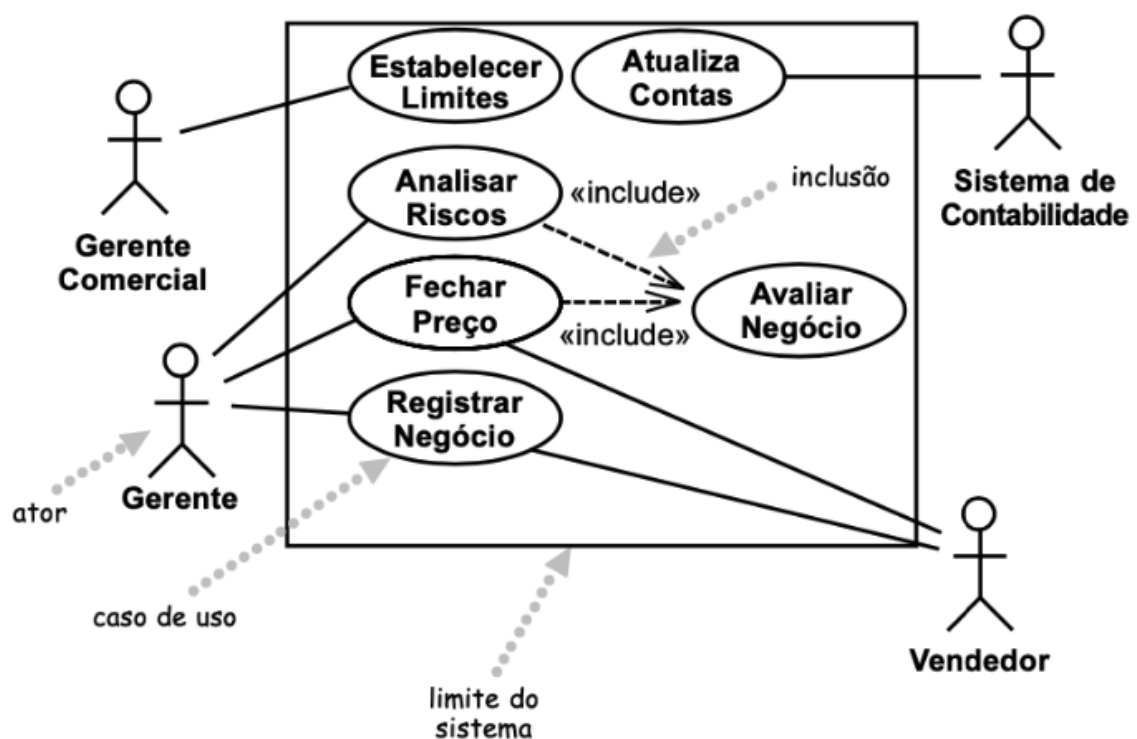


Figura 1. Exemplo de Diagrama de Caso de Uso (Retirado do Livro UML Essencial do Martin Fowler).

Modelo de Domínio

Confeccionar a primeira versão do Modelo de Domínio identificando as classes conceituais relacionadas com os requisitos e modelando os atributos e associações adequadas. Consultar as referências bibliográfica no plano de ensino para exemplos concretos.

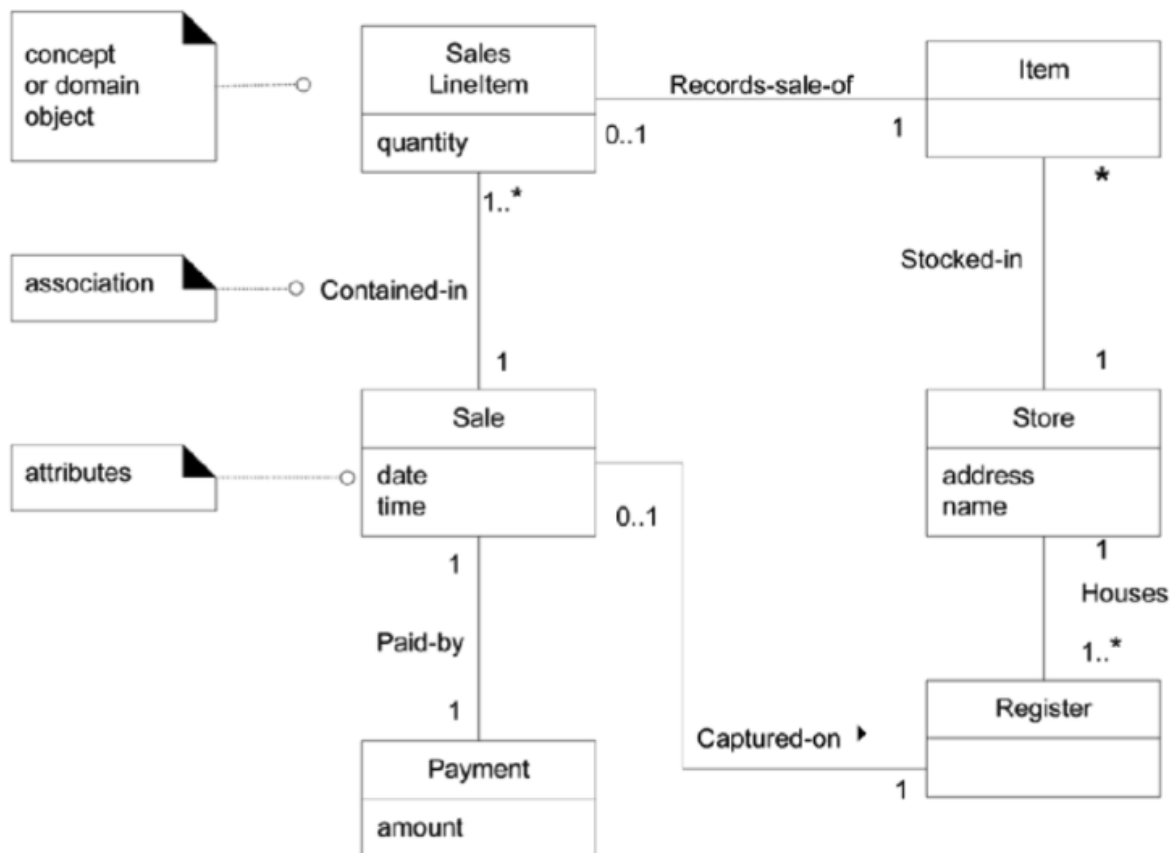


Figura 2. Exemplo de Modelo de Domínio (Retirado do Livro Utilizando UML e Padrões do Craig Larman).

Especificação de Caso de Uso

Escolha o caso de uso considere mais arquiteturalmente significativo associado ao escopo que lhe foi atribuído e, para este caso de uso, confeccione o documento Especificação de Caso de Uso. A especificação deve conter as seções Título, Descrição, Pré-condições, Fluxos de Eventos, Pós-Condições e Regras de Negócio Específicas. A seção Fluxo de Eventos conter ao menos o Fluxo Básico e, caso necessário, também deve conter Subfluxos, Fluxos Alternativos e Fluxos de Exceção, os quais devem apresentar cada passo numerado. As Regras de Negócio Específicas devem ser numeradas e referenciadas em um ou mais fluxos. Caso uma mesma regra de negócio específica seja mencionada em mais de um caso de uso, essa regra deve virar uma regra de negócio geral em um documento a parte. Consultar as referências bibliográfica no plano de ensino para exemplos concretos. Apenas serão aceitas especificações de caso de uso confeccionadas utilizando a linguagem de marcação de texto Markdown no repositório do projeto.

Caso de uso CDU1: processar venda

Escopo: Aplicação PDV ProxGer

Nível: objetivo do usuário

Ator Principal: Caixa

Interessados e Interesses:

- Caixa: deseja entrada de pagamento rápida, precisa e sem erros, pois a falta de dinheiro na gaveta do caixa será deduzida do seu salário.
- Vendedor: deseja comissões sobre vendas atualizadas.
- Cliente: deseja comprar, receber um serviço rápido e com o mínimo esforço. Deseja a exibição, facilmente visível, dos itens e preços inseridos. Deseja um comprovante da compra, necessário no caso de devoluções de mercadorias.
- Empresa: deseja registrar precisamente as transações e satisfazer aos interesses do cliente. Quer garantir que os pagamentos a receber do Serviço de Autorização de Pagamentos sejam registrados. Deseja algum tipo de proteção contra falhas para permitir que as vendas sejam capturadas mesmo se os componentes do servidor (por exemplo, validação remota de crédito) se encontrarem indisponíveis. Deseja uma atualização automática e rápida da contabilidade e do estoque.
- Gerente: deseja poder realizar rapidamente operações de correção e facilmente corrigir os problemas do Caixa.

Figura 3. Exemplo de Especificação de Caso de Uso (Retirado do Livro Utilizando UML e Padrões do Craig Larman).

Contrato de operação

Considerando a **Especificação de Caso Uso** *elaborada*, selecionar, dentre algum dos fluxos (básicos, alternativos, sub-fluxos), o **contrato de operação** mais arquiteturalmente significativo e, para este contrato de operação especificar:

1. as **pré-condições** ; e
 - a. existência de objetos
 - b. existência de associações entre objetos
 - c. e existência de objetos com estado específico
2. as **pós-condições** com as respectivas indicações
 - a. Criação ou exclusão de instância de objetos;
 - b. Criação ou exclusão de associações entre objetos;
 - c. Modificações de valores dos atributos;

Contrato CO2: entrarItem

Operação:	entrarItem (idItem: IdItem, qtdade: inteiro)
Referências Cruzadas:	Casos de Uso: Processar Venda
Pré-Condições:	Existe uma venda em andamento.
Pós-Condições:	<ul style="list-style-type: none">– Foi criada uma instância liv de LinhaDeItemDeVenda (<i>criação de instância</i>).– liv foi associada com a Venda corrente (<i>associação formada</i>).– liv.qtdade tornou-se quantidade (<i>modificação de atributo</i>).– liv foi associada a uma EspecificaçãoDeProduto, com base na correspondência de idItem (<i>associação formada</i>).

Figura 1. Exemplo de Contrato de Operação (Retirado do Livro Utilizando UML e Padrões do Craig Larman)

Diagrama de Colaboração

Para cada o contrato de operação selecionado, apresentar:

1. O **projeto de objetos para a pré-condição** especificada no contrato de operação;
2. O **projeto de objetos para pós-condição** especificada no contrato de operação;
3. O **diagrama de comunicação** para a operação de sistema detalhando as interações que transformam o conjunto de objetos na pré-condição no conjunto de objetos na pós-condição.

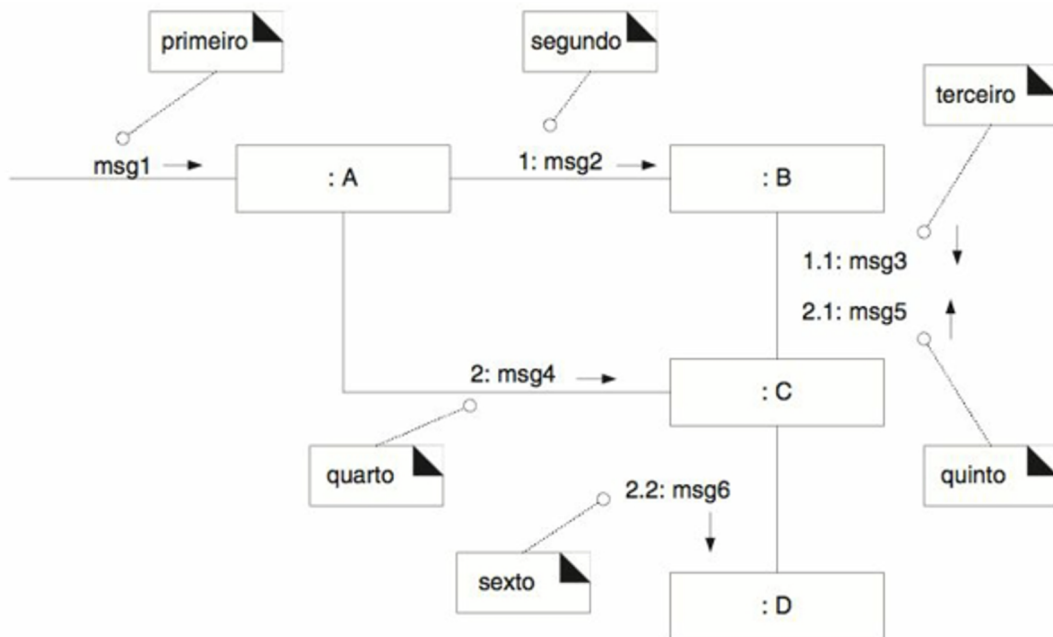


Figura 2. Exemplo de Diagrama de Colaboração (Retirado do Livro Utilizando UML e Padrões do Craig Larman)

Diagrama de Classe

Confeccionar o Diagrama de Classes contendo as classes de projeto e as respectivas operações necessárias a partir das trocas de mensagens entre os objetos no Diagrama de Colaboração confeccionado.

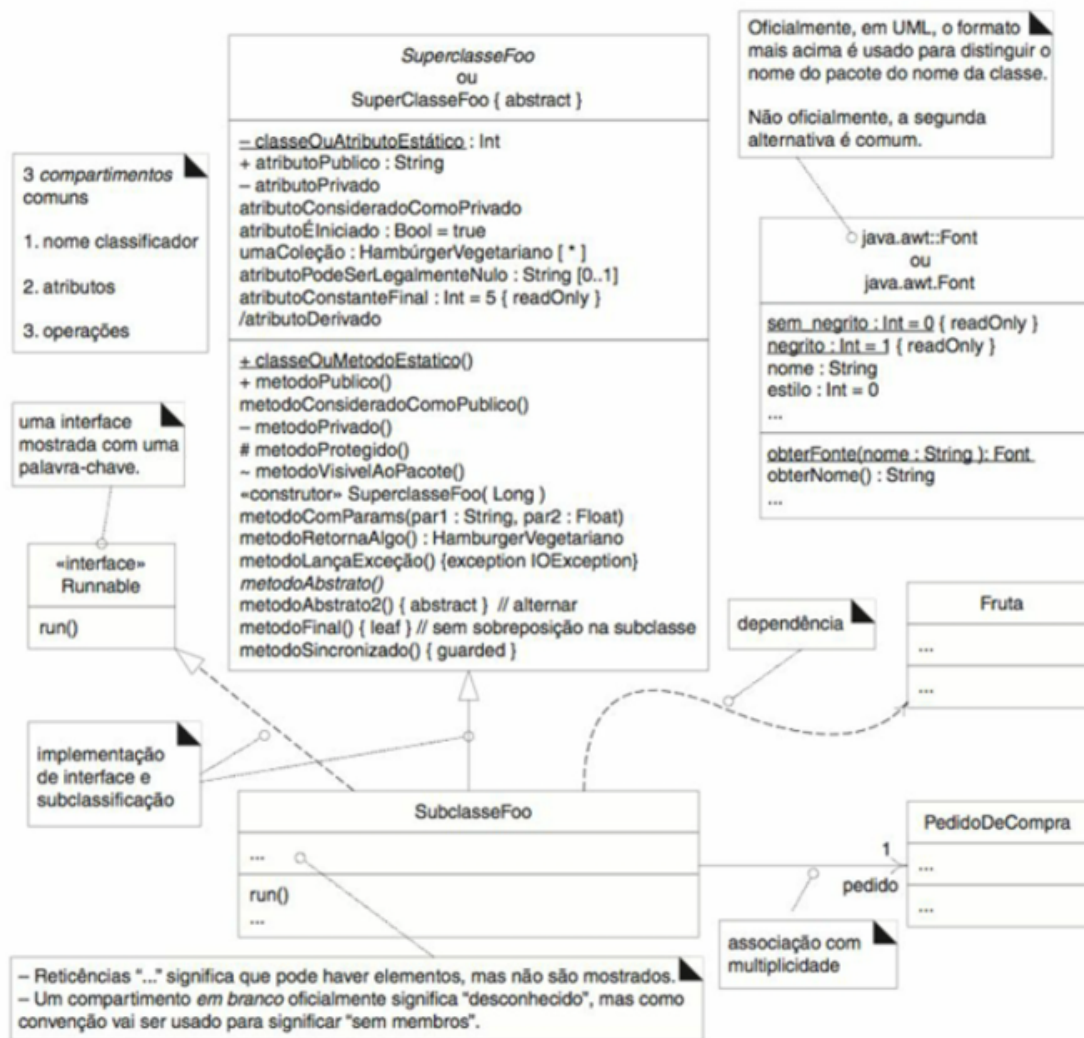


Figura 16.1 Notação comum de diagrama de classe UML.

Figura 3. Exemplo de Diagrama de Classes (Retirado do Livro Utilizando UML e Padrões do Craig Larman)

Sprint 2

O objetivo da Sprint é finalizar a elaboração utilizando a metodologia Test-Driven Development (TDD) para implementar testes automatizados e o respectivo código com padrões de projeto que para os testes passarem. Para isso, vamos utilizar contratos de operação confeccionados. A pré-condição do contrato de operação será o contexto do teste, também chamado de **fixture**. A pós-condição do contrato de operação deveser verificada utilizando comandos do tipo **assert**.

Nesta sprint, o artefatos esperado é **código-fonte utilizando a linguagem de programação Java**. Para isso, a implementação deve constar

- Classes de testes na pasta *src/test* contendo **Casos de Teste (Test Case)**
- Classes de implementação na pasta *src/test* contendo a implementação com padrões de projeto que fazem os testes passarem.

Utilizar as atividades assíncronas como exemplos para as classes de testes e classes de implementação.

É esperado que existam **um mínimo de casos de testes não-triviais**.